Title: Math Club

Author: Harry Miller

Course: BSC (HONS) COMPUTING

Project code: PJE40

Supervisor: Amanda Peart

Month Year: May 2025

Please tick

- ☑ I give permission for my project to be published in the University library and/or be made available to other students as examples of previous work. (optional).
- ☑ I confirm that I have read and understood the University Rules in respect of plagiarism and student misconduct.
- ☑ I declare that this work is entirely my own. Each quotation or contribution cited from other work is fully referenced.

Date: 02/05/2025

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Amanda Peart for her guidance, expertise, and continuous support throughout this project.

Abstract

This project introduces Math Club, a web application designed to improve primary school students' mathematical skills through daily engagement. The app targets children in years 1-6. The app employs a "little and often" approach with randomly generated, age-appropriate questions to develop consistent habits while building confidence in mathematical concepts.

Research demonstrates positive impacts of technology in mathematics education. UCL studies found 90% of reviewed math apps improved learning outcomes among primary students. Department for Education data showed stable mathematics attainment post pandemic. Research indicated that apps offering personalised learning with immediate feedback significantly enhanced student engagement and confidence with mathematical concepts.

The design process focused on creating a child-friendly interface with high usability. I did this by employing: bright colors, simple interfaces with only necessary options, simple navigation, clear child friendly texts and more. Multiple design iterations, notes and diagrams defined the application during early stages of development.

The requirements took influence from the designs and literature review

I developed the app using Flutter on Visual Studio Code, with Firebase for backend services. Key features include randomised question generation tailored to the school year and consecutive completion tracking. I implemented user authentication, profile management, question algorithms, progress tracking and more.

Testing validated functionality, performance, compatibility, usability, correctness and grammatical correctness, with 92 coded tests conducted. Testing identified improvements needed before deployment but was overall positive.

After reviewing my requirements and concluding my project (this doesn’t mean I am done developing), I got a better understanding of if/how I reached my requirements and what had to be done before deployment.

Table of contents

Table of figures (table colour change on chapter change and some chapters don't have any figures)

Chapter 1: Introduction

1.1. Math Club

My app is a desktop maths app called math club. The primary purpose of the app is to generate random questions for each primary school age group. The app employs a "little and often" approach to develop consistent habits while building confidence in mathematical concepts. Each age group would have increasingly difficult possible questions as number ranges get increased and more factors get included. Random questions would be generated everyday and daily consecutive completion would be stored to encourage the ‘little and often’ approach.

The reason I chose to make a math app is because they significantly increase learning outcomes in primary school children. Research demonstrates positive impacts of technology in mathematics education. UCL studies found 90% of reviewed math apps improved learning outcomes among primary students (Maths Apps: The Pros and Cons, 2023). Research indicated that apps offering personalised learning with immediate feedback significantly enhanced student engagement and confidence with mathematical concepts. My app could help children realise their academic potential and achieve higher grades.

1.2. Target Audience

This app is designed for primary school children (children in years 1-6), ensuring accessibility on different platforms so all students can participate. Its design is child-friendly, incorporating engaging colours and images, simple navigation, and clear text to maximise usability. Some pages are tailored for parents and teachers, providing necessary information about the app through text-based content. (Appendix A: project initiation document, 1. Target Audience)

1.3. Project aim and objectives

Aim: The aim for this project was to produce a maths application that incentivises children to do more maths little and often. The app should help children build a habit out of doing maths

Objectives:

- Help children who would prefer to use computers than paper
- Help children enjoy maths more
- Help children learn and revise maths fundamentals
- Help children achieve higher grades in maths
- Help children put more overall time into there maths studies
- Give teachers and parents more productive computer tasks for children

Development Objectives;

- Completing pid and ethics form and getting them approved
- Literature review primary school maths
- Evaluate maths applications
- Determine requirements
- Design application
- Implement application based off design
- Test functionality and usability of application
- Evaluate project and write report

- Prepare for presentation

A personal goal of this project for me was to develop an application that comes as close to industry standards as I am currently capable of developing.

1.4. Project constraints

The constraints faced in this project included working independently with limited knowledge and resources, which required skill development and careful planning.
Time/deadlines was also an issue as this was a very large project and needed to be done whilst also completing other coursework. Time management and adaptability were crucial to overcome these challenges and stay on track. (Appendix A: project initiation document, 5. Project constraints)

1.5. Log of risks

No	Description	Likelihood (high, medium, low)	Impact	Mitigation/Avoidance
1	Laptop failure	Medium	Cause delays to the project timeline and data loss	Back up work to a USB and another source
2	Breaking laws regarding data, childrens data and security	Low	Possible legal situation	Research what i can and can't do
4	Missing deadlines	High	Capped or fail module	Stay on top of my work and plan ahead and with extra time
5	App not being usable regarding functionality	High	Low grade	Make sure i have enough time to program all features
6	App not being usable regarding GUI	Low	Low	Put time into making the app look nice
7	Not being able to program backend elements	High	Low grade	Make sure i have enough time to program all features
8	No time to test	High	Low grade	Leave time to do detailed tests
9	Inadequate research	High	-Low grade -Break laws	Make sure I get enough reliable research

			-unusable app	
10	App not appealing to target audience	High	Waste of time	Try my best to make the app appeal to the target audience
11	Lack of feedback	Medium	App developed from one perspective	Get as much feedback as i can from my project supervisor
12	Project too big	Medium	-unfinished app -low grade	Leave enough time to make it
13	Myself	medium	-unfinished app - low grade	- Use slippage time

figure 1.1. Log of risks

1.6. Project Management

My project will require a structured approach with effective time management and milestone planning. I'll divide the work into key phases including project initiation, research, design, implementation, and testing so the project is more approachable. Regular check-ins with my supervisor throughout the development process will provide valuable feedback and guidance.

My risk management plan identifies thirteen specific risks, including technical issues, compliance concerns, and personal factors.I've established mitigation strategies like maintaining backup systems and allowing slippage time in my schedule.

While I may need to adapt my timeline as the project progresses, I aim to maintain a methodical progression through each phase. This approach should allow for effective management of this complex project despite its challenges.

1.7. Legal, ethical, professional, social issues

My project followed legal, ethical, professional, and social guidelines.

Legal:
During my development journey, I complied with the GDPR and the university guidelines. When my application is ready for deployment, I will however, need to make sure that I can collect and handle data lawfully.

Ethical:
I used firebase as it offered automatic encryption and password hashing. This increased the app's security with little effort.
My application doesn’t reinforce any biases or stereotypes.

Professional:
I have created a safe learning environment with exclusively accurate information
The interface and architectural design is user centered specifically to my young target audience. Usability was a big concern throughout design, implementation and testing.

Social:
To ensure accessibility I made the decision during design that my app should be a web app; this is due to not all primary school children having a mobile phone.(Appendix A: project initiation document, 12. Legal, ethical, professional, social issues)

I had to apply for an ethics certificate as part of complying with requirements. My ethics were reviewed by the university and my supervisor. For my ethics certificate (see Appendix B: Ethics certificate)

1.8. Project Approach

My project required a structured approach with effective time management and milestone planning. I anticipated obstacles so tried my best to be ahead, and maintained regular check-ins with my supervisor throughout.

I made my app on visual studio code using Flutter/Dart as my primary language. My question generation algorithm was designed to align with age appropriate (school years 1 to 6) difficulty levels using the national curriculum for guidance. (Appendix A: project initiation document, 9. Project approach)

1.9. Project deliverables

My project deliverables are:
-math club app
-report
-project initiation document
-Ethics certificate

1.10. The report

This report provides a comprehensive overview of the Math Club app development process. Chapter 2 presents a literature review examining current math education approaches and similar applications. Chapter 3 details the design process. Chapter 4 defines functional and non-functional requirements. Chapter 5 covers the implementation process including problems encountered and changes from design. Chapter 6 documents testing methodologies and results. Chapter 7 evaluates the project on its original requirements and documents what's left to complete. Chapter 8 concludes the project.

Chapter 2: Literature Review on Maths Apps in Primary Education

2.1 Introduction

Technology is becoming an important part of education, especially for young learners. Math, a subject that can be tough for many children, has seen the rise of apps designed to make learning easier and more fun. This review will look at how children in primary school interact with math apps, whether these apps are helpful, and if they improve grades.

The integration of math apps into primary school curriculum reflects a broader shift towards embracing technology to foster personalised and engaging learning environments. Math apps have been designed to cater to the needs of young learners. However, questions remain about their actual effectiveness: Do these apps improve students' grades? How do they influence a child's motivation and attitude toward mathematics? And, critically, do they align with children's natural inclinations for how they like to study and explore mathematical concepts?

Understanding how children prefer to study math is a crucial aspect of this review. Different learners have varying preferences, some thrive in structured environments, while others excel through playful exploration. Investigating whether math apps align with these preferences will provide valuable insights into their suitability as a teaching tool.

By analysing existing research this literature review seeks to offer a comprehensive understanding of mathematics apps' role in primary education. The goal is to determine whether these apps not only serve as a supplement to traditional teaching methods but also transform how young learners perceive and engage with mathematics.

2.2 Research

2.2.1. Are primary school maths grades improving?

In 2023, due to the pandemic, it was expected that there would be lower levels of attainment in maths, while children caught up on lost learning. Results showed that 73% of pupils met the expected standard in maths, up from 71% in 2022 (DfE, 2023). In 2024, 73% of pupils met the expected standard, unchanged since 2023 (DfE, 2025). These results show that even directly after the pandemic, math levels have improved.

2.2.2. Do primary school aged children like to do maths?

No, not all children like maths.

Some children don't see how learning about math in the classroom can be applied in the real world. Also there is a general attitude of people towards math is that it is altogether difficult and stressful. Speaking negatively about it discourages children. It might stop them from trying their best or even trying at all. Also children often link their negative feelings about math to past bad experiences. They don't suddenly start disliking the subject; this dislike

develops gradually over time through repeated struggles or unpleasant encounters with math. (Segundo, 2023)

2.2.3. Do math apps improve math skills/grades?

Maths apps are convenient, as they can be downloaded onto devices, making them accessible to children anytime and anywhere. These apps allow children to progress at their own pace, eliminating the pressure of keeping up with classroom schedules and providing more time to complete activities. This flexibility is especially beneficial for students who may need extra practice or support.

Researchers from University College London (UCL) conducted a study analysing over 70 maths apps developed in the past three years, including the 25 highest-rated ones. Their findings showed that 90% of these apps contributed positively to improving children's mathematical learning and development (Maths Apps: The Pros and Cons, 2023). This demonstrates the potential of technology to complement traditional teaching methods.

The research further highlighted that apps offering a personalised learning journey were particularly effective in maximising learning outcomes. These apps provide children with immediate feedback on their answers, indicating whether they are correct or incorrect. They also track progress, which helps educators and parents monitor development, and reward achievements, creating a sense of accomplishment and boosting motivation.

Additionally, many apps incorporate engaging features such as games, challenges, and interactive visuals, which make learning more enjoyable for children. These elements not only enhance interest in mathematics but also help reduce anxiety and build confidence. By fostering a positive attitude toward math, these apps encourage students to approach the subject with enthusiasm.

In 2019, the Department for Education (DfE) outlined its vision to assist schools and colleges in effectively integrating technology (DfE, 2021). This initiative aimed to help educational institutions better understand the opportunities technology offers and address the barriers to successfully adopting and utilising EdTech (Educational Technology). To support its EdTech Strategy and guide future policy development, the DfE enlisted CooperGibson Research (CGR) to conduct a study assessing the use and state of technology across schools in England.

In March 2020, the Department for Education initiated the commissioning process for a survey to explore the EdTech landscape within schools (DfE, 2021). The primary purpose of this survey was to gain insights into the current state of technology adoption and use. This information would help inform government actions to assist schools in integrating technology effectively, achieving cost savings, reducing workloads, and improving student outcomes. Additionally, the survey aimed to help the EdTech sector understand the needs of schools, allowing developers to adapt and create tools that align with the specific conditions and requirements of the school environment.

Bellow are the key findings taken directly from the report (DfE, 2021)

Benefits of EdTech
Perceptions of the impact of technology on pupil attainment were positive:
• The majority of headteachers (88%) and teachers (84%) indicated that technology had or would contribute to improved pupil attainment. Over half believed that this positive impact had already been experienced (headteachers 55%, teachers 57%).
• The majority of headteachers (74%) and teachers (65%) also indicated that technology already had, or would in the future, contribute to reduced workload.
• Headteachers believed that the use of technology had saved them time on key tasks, most commonly for financial management, engaging with parents and governance, plus pupil data management and timetabling for secondary schools.
• Teachers were also broadly positive about the impact of technology on the time taken to complete teaching-related tasks, especially for collaborating and sharing resources with other teachers, tracking pupil progress, planning lessons or curriculum content, and delivering lessons(DfE, 2021)

The report carried out showed that the majority of schools believed that they could effectively support pupils to learn remotely using apps on tablets and mobiles etc. The teachers indicated that the new technology would help them to deliver better remote education in the future and many also believed it would help them to deliver better in-class education.

The study indicates that apps have a positive impact on young learners, improving their educational results.

2.2.4. How do children of primary school age interact with math apps?

Maths apps are increasingly being used both in schools and at home to improve children's learning experiences. These apps have gained significant popularity due to their interactive features, engaging content, and ability to provide tailored learning experiences for students of various skill levels. With over 70% of children now having access to touch-screen devices, the use of these tools has become widespread and accessible, making them an integral part of modern education (DfE, 2021).

41% of teachers in early primary schools are incorporating maths apps into their teaching practices. These apps are used as supplementary tools to complement traditional methods.

Furthermore, these apps are not limited to school environments, they are also widely used at home, allowing children to practice independently and at their own pace. Parents can turn to maths apps to support their children's learning, providing additional opportunities for practice outside the classroom.

2.2.5. Usability

A well-designed app must suit its target audience, considering age and accessibility for an intuitive experience.

Nielsen's heuristics highlighted the importance of system status visibility, user control, and consistency. These are all fundamental aspects of effective app design. Providing immediate feedback helps users track their progress, while intuitive navigation reduces frustration. Maintaining uniform terminology and interface elements enhances accessibility, ensuring a seamless user experience. (Nielsen, 1994)

Schneiderman's Golden Rules focused on structured workflows, error prevention, and intuitive interaction. He found predictable navigation ensured ease of use, especially for young learners. Informative feedback deepens understanding, and well-designed input mechanisms reduce accidental errors. (Wong, 2025)

These usability principles should guide every app's development, ensuring a balance between accessibility, engagement, and effective learning.

2.3. Review of current math apps

As part of this Literature review, I have looked at a variety of maths apps to try and ascertain what I felt were the good features and those that need improving. Below are two of the reviews carried out with “CGP Books” and “Math Games”.

2.3.1. CGP BOOKS (Bushell & Greenway & Palin & Purvis Simpson)

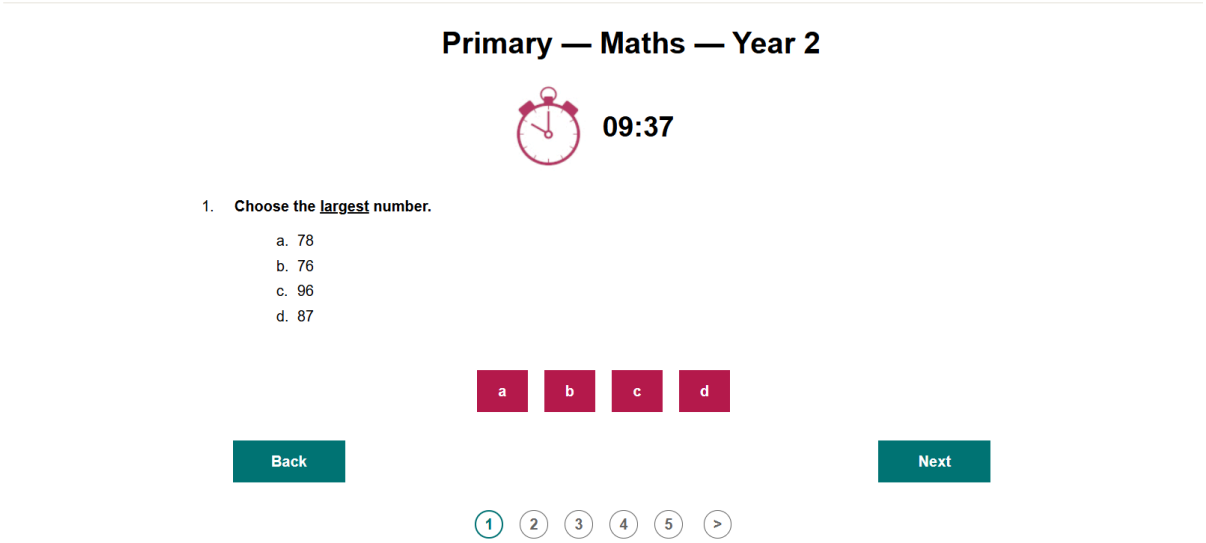


figure 2.1. Screenshot of CGP Books interface

The app lacks intuitive navigation, making it difficult for younger users to find their way around. The interface requires multiple clicks to proceed, and feedback on answers is delayed, reducing engagement. Graphics are minimal, and questions aren't randomised, which limits variety. However, it does allow school year selection and maintains an appropriate difficulty level. The "About Us" page is well-structured, and account creation appears simple and secure. The terminology used, such as “Infrequently Asked Questions,” can be confusing, and the site seems overly focused on sales rather than usability. These findings highlighted areas for improvement in user experience, engagement, and personalisation while recognising its strengths in security and content organisation. I took these findings and used them in my app design.

2.3.2. **MATH GAMES** (Practice Math Games)

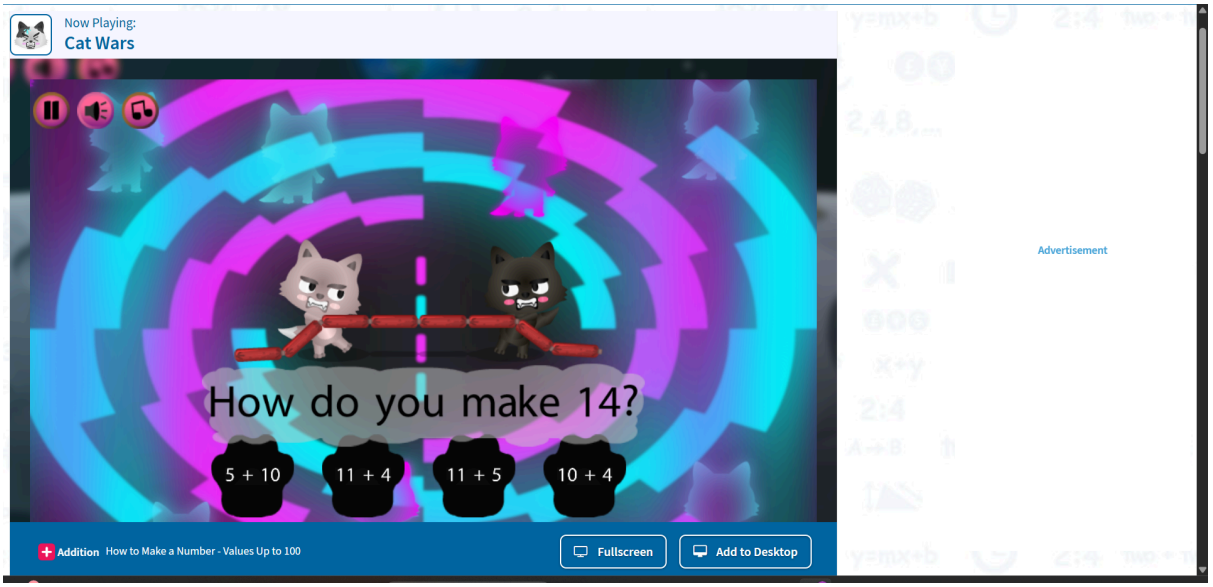


figure 2.2. Screenshot of Math Games interface

This app offers only specific quizzes, lacks randomisation, and repeats questions, making it dull and predictable. While it provides instant feedback on answers, it does not store previous scores, limiting users' ability to track progress over time. Navigation is unintuitive, making it difficult for younger children to use independently, and the interface lacks a back button, forcing users to rely on the browser to exit or navigate backward. School year selection is available but must be re-entered for each session, adding unnecessary friction. There is no advanced mode and no About Us page. Additionally, the FAQ section contains excessive information, making it hard to find relevant details quickly. The app appears more suited to classroom environments rather than individual use, as younger children may struggle with its complexity. However, the account creation process is straightforward and secure, offering a positive feature. These findings highlighted areas for improvement that should focus on usability, engagement, and personalisation to create a more accessible and enjoyable learning experience. Its main strengths were account creation and security. I took these findings and used them in my app design.

2.3.3. Conclusion of website reviews

I initially planned to use a set of predetermined questions to ascertain the quality of the apps, but I soon realised that this would not work as the designs and usability varied so much. I did try and carry out the same functions on each app where possible, for example, looking at the FAQ page. I found that there were large differences in every app I looked at. Some performed some tasks well and failed miserably at others, for example, CGP BOOKS let you select a school year but didn’t store previous results. The key to the review was to harvest all the good points in the other apps and eliminate the poor in my design.

What makes a good maths app?

My research (2.3. Review of current math apps) suggests the following helps to make a good maths app for primary school children.

Tracking Progress. Apps that allow for the progress to be tracked assist parents and teachers in monitoring development. This will identify when additional support is needed.

Accessibility. The more compatible with different devices the better. This allows children to learn on the go and can make it easier to integrate into their daily routine.

Motivation. When a child is motivated, they are more likely to enjoy the work. This can be done by offering words of encouragement, providing feedback and offering rewards.

Repetition. Regular practice will help to reinforce the different mathematical concepts. This will help a child understand and remember.

Age/Ability Appropriate. It is important that an app provides age and/or ability related work. Following the national curriculum will ensure that the app is age appropriate. Having advanced maths features will help ensure children with greater ability do not become bored.

Usability. A good app must know its target audience. The design must focus on how the app will be used considering the users age and other abilities. From the apps I looked at I would say the usability varied quite significantly. The usability of my app is further on in this report but was heavily based on my reviews carried out on current available apps. You can see from the above reviews on “CGP Books” and “Math Games”, that my focus was on usability as much as content.

2.4. Conclusion

The integration of maths apps into primary education has shown promising benefits for young learners. Research suggests that these apps not only provide accessibility and flexibility but also support individualised learning experiences that cater to different learning styles.

Although not all children enjoy mathematics, gamified learning and personalised feedback appear to significantly improve their attitude and confidence in the subject. The findings from University College London’s study (Maths Apps: The Pros and Cons, 2023) confirm that most math apps enhance mathematical understanding, demonstrating their potential as effective educational tools.

Furthermore, government-led research (DfE, 2021) highlights the broader benefits of educational EdTech, such as reduced teacher workload and improved lesson delivery. As schools continue to embrace technology, math apps will likely play an increasingly vital role in shaping student experiences and learning outcomes.

As this review highlights, not all math apps are equally effective. While some provide structured learning aligned with the school curriculum, usability challenges, repetitive content, and limited feedback mechanisms can reduce engagement and effectiveness. Apps that track progress, motivate learners, ensure accessibility across devices, and offer age-appropriate challenges stand out as key factors for success.

By identifying both strong and weak features in existing math apps, this review underscores the importance of thoughtful design in educational technology. Effective apps must prioritise usability, engagement, and alignment with learning objectives to truly support young learners in developing math skills.

Maths apps can make learning enjoyable by using games, puzzles, and challenges to keep children interested and motivated. By turning learning into fun they are more likely to engage and retain information. By practising math on their own, children may experience reduced fear and anxiety that can accompany the subject. Through consistent use, math apps create an environment where children feel supported and motivated to explore math concepts without feeling pressured or overwhelmed.

A well-designed app must suit its target audience, considering age and accessibility. Nielsen’s heuristics highlighted the importance of system status visibility, user control, and consistency which are fundamental aspects of effective app design. Maintaining uniform terminology and interface elements enhances accessibility, ensuring a seamless user experience. Schneiderman’s Golden Rules focused on structured workflows, error prevention, and intuitive interaction. He focused on predictable navigation, informative feedback and well-designed input mechanisms.These usability principles should guide every app’s development.

Chapter 3: Design

3.1. Introduction

Due to an unorthodox order of development parts of the design chapter were performed before I had finished defining my requirements. This chapter played a huge part in defining all aspects of my app during early stages of development. Completing this chapter also gave me a lot of ideas that would later be implemented such as the majority of the app architecture (some features were added after design) , navigation layout, user interface (ui) design and more.

3.2. App architecture

-Register page
Input - username, school year, password
Store - account ID, username, school year, password hash
Firebase database
If valid details sends user to homepage/questions

-login page
Input - username, password
Check credentials with firebase
If valid details sends user to homepage/questions

-navigation
Navigate to question page after logging in
drop down menu with links to all pages at top
Name of current page next to menu button

-Questions page
10 daily questions based on school year
+1 after correct answer
Show answer if incorrect, does not count towards consecutive completion
click anywhere to continue after a question regardless if correct
Consecutive completion stored and displayed on question and account pages
-can continue after completing all questions

-Random questions generated
Questions will only be somewhat random- they will be random out of all the possible options but ranges set to minimise difficulty will limit the app from being truly random.

Different conditions for users in different school years
Each age will have multiple types of questions
A type of question could be multiplication, division, subtraction, addition, or more trivial questions like shapes
A operation may have multiple types of questions such as high*low or medium*medium for certain ages
Each type of questions will have its own conditions such as ranges, operations, possible answers, multiple choice or type answer
Different types of questions will use different stored text values
Number ranges will be different for different operations

-Ranges for 1st and 2nd numbers may be different for some questions.
-No remainders
-more questions at later ages
-possible question types with more complex operations and 3 numbers for later ages
-possible questions that require user to type at later ages

-False answers
False answers will also need to be randomly generated, this could be done by randomly applying addition subtraction operations to correct answer
For trivial questions the incorrect answers would be the correct answer for a different question of the same type. For example a different shape

-images
Questions will be displayed with an image, some questions may call specific images, others may be random from a selection of possible images

-Example of questions for younger ages(pre research just ideas)
2 x 12
10 / 5
4 + 8
16 - 7
What shape is this
How many sides does a square have

-Store time
add 1 to users year every july when they break up so they get questions based on following year

-Account page
Show: username, highest consecutive completion, current consecutive completion, learning age
Automatically deleting accounts due to data restrictions after it becomes inactive?

-Advanced mode
+1 to users current school year
Extra stage added for year 6 with + 1
Toggle on the account page. Displayed as on with + 1 next to leaning age

-About us
Paragraph
Contact details
-Faq
searchable
Questions about app
How can this help
How can we contact- link about page

-Algorithms
Randomly generate:
Question type, operation, 1st and 2nd number, or one of the options of questions for trivial types, incorrect answers
Correct answers would be calculated after the sum is generated or for trivial questions it will be stored in the program with the question.
Consecutive completion will be calculated and stored

Highest consecutive completion will be stored and updated if it’s lower than current consecutive completion
Learning age and other account values stored

3.3. Visual representations

3.3.1. Architecture design

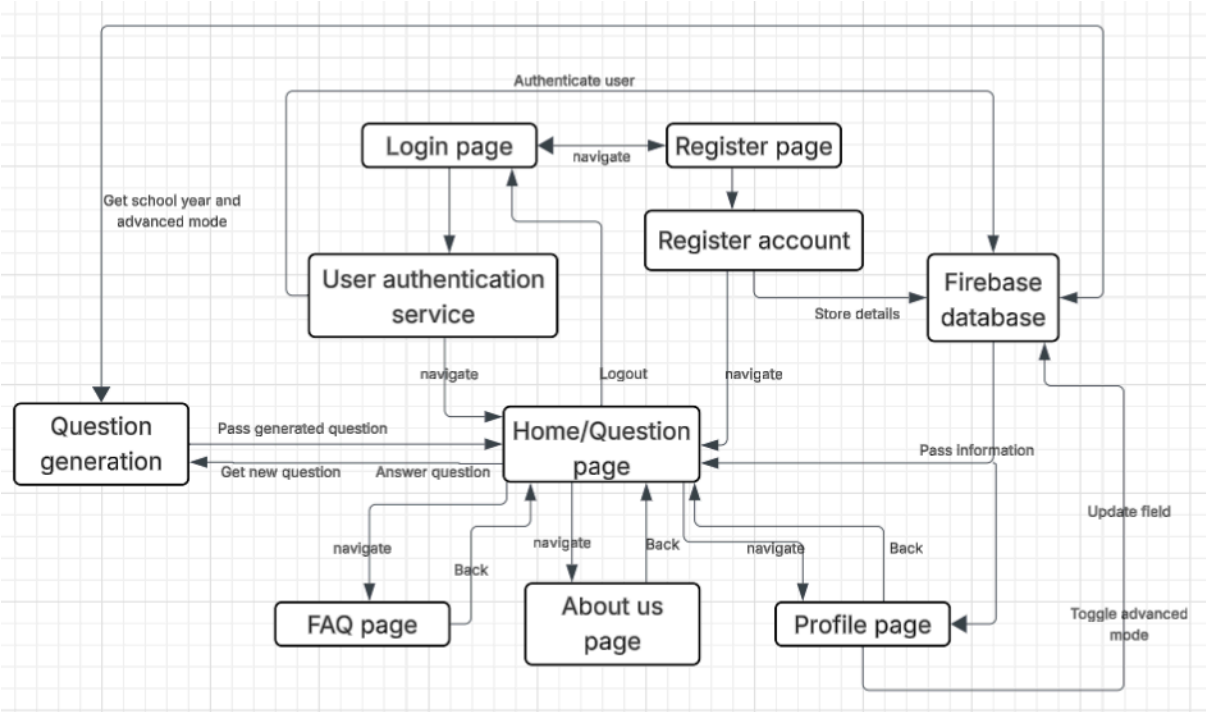


figure 3.1. Architecture design

3.3.2. Use cases

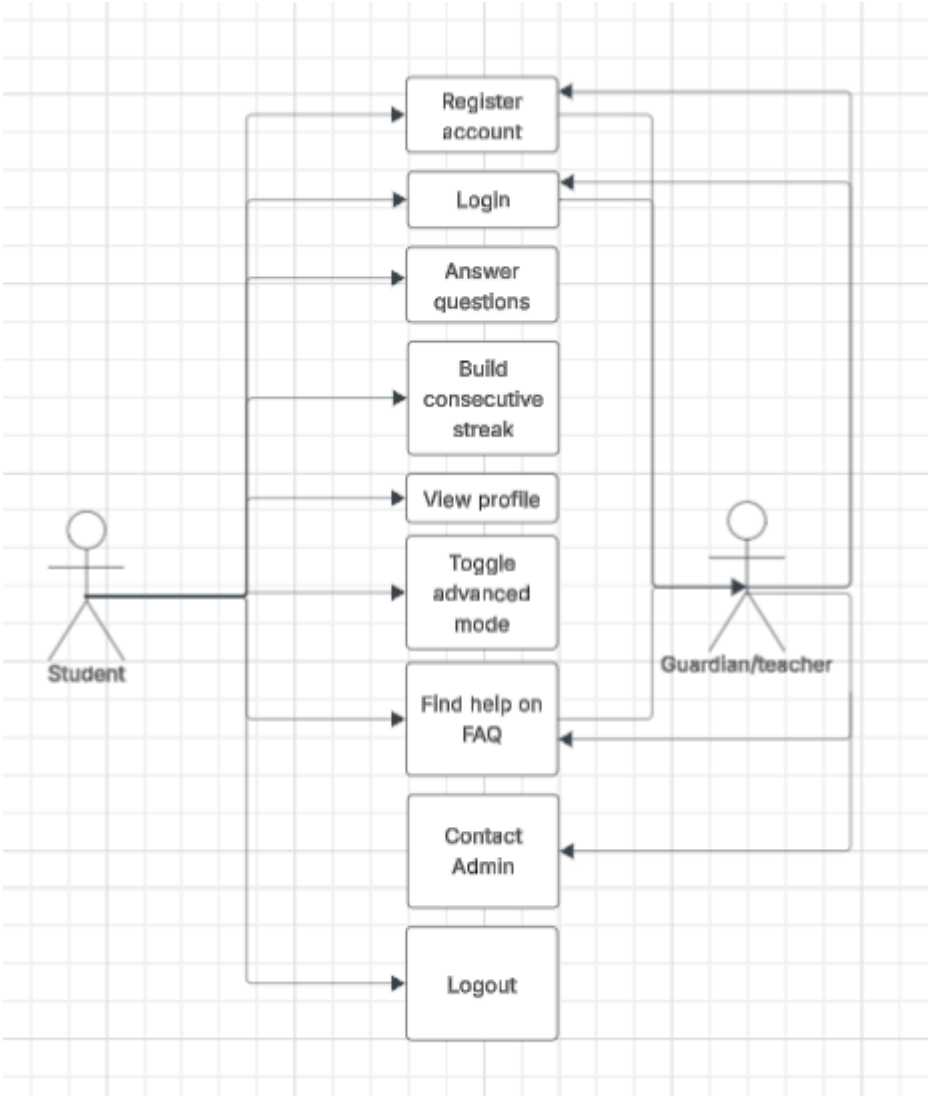


figure 3.2. Use cases

Arrows going from student or parent/guardian represent them using the application.
Arrows going to guardian/teacher represent students asking for help.

3.3.2. Entity relationship diagram of firebase database

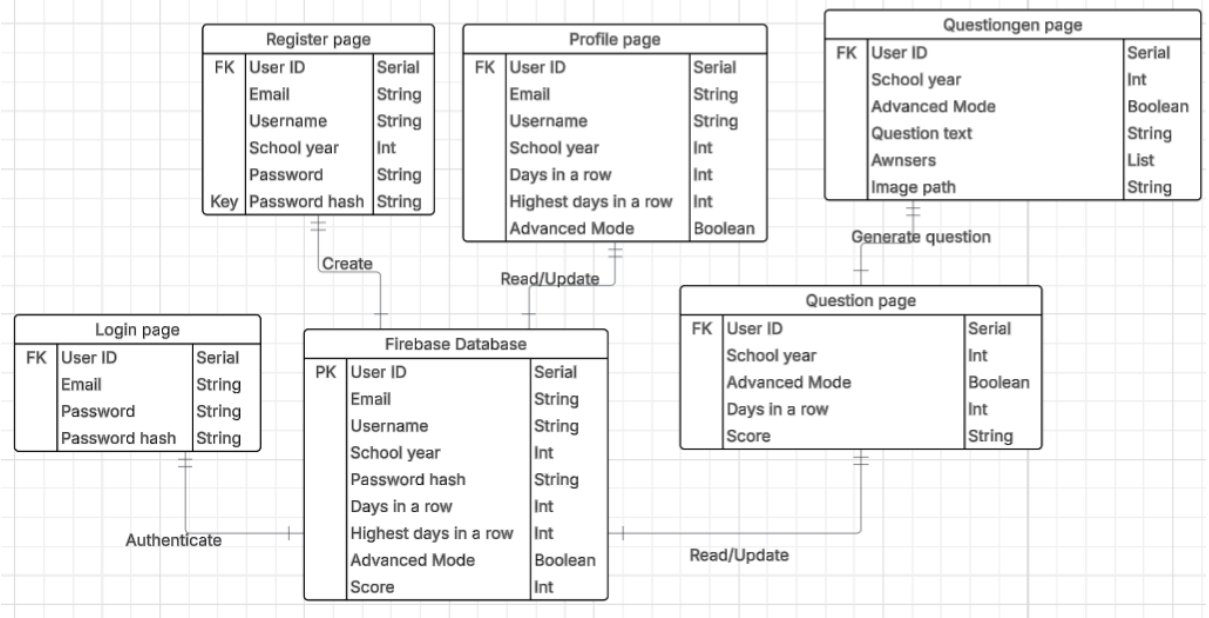


figure 3.3. Entity relationship diagram of firebase database

This table shows how different pages and features will use the firebase database. Login page will authenticate users' email and password. It will access users stored password hash to authenticate
Register creates an account and all the values are stored to firebase upon successful registration.
From profile page users can read their details or toggle advanced mode
From the question page users can see their score out of 10 and days in a row. Users can update these values by completing questions. Advanced mode and school year are passed into question gen
Questiongen uses school year and advanced mode and combines these for the difficulty level. It generates questions at this difficulty and passes question text, answers and image path back to question page

3.4. Areas of research needed to make app

Implementation research
Research quick maths questions for school years 1-6 and advanced year 6
-Research advanced year 6 maths for students using advanced mode in year 6.
-What questions are appropriate for each learning age?
-Use this to create my own questions in implementation,

Laws, rules and regulations
Data laws regarding children and data
-What data can I lawfully store?

-Do i need to automatically delete data when the users have passed year 6 or the account is inactive?
-Laws/rules regarding testing depending on chosen testing methods.

University rules that may affect my project

-Possible rules and regulations, most likely to affect testing.

3.5. Designs

3.5.1. First designs

Many key features are missing. These designs are basic and early in early stages of development.

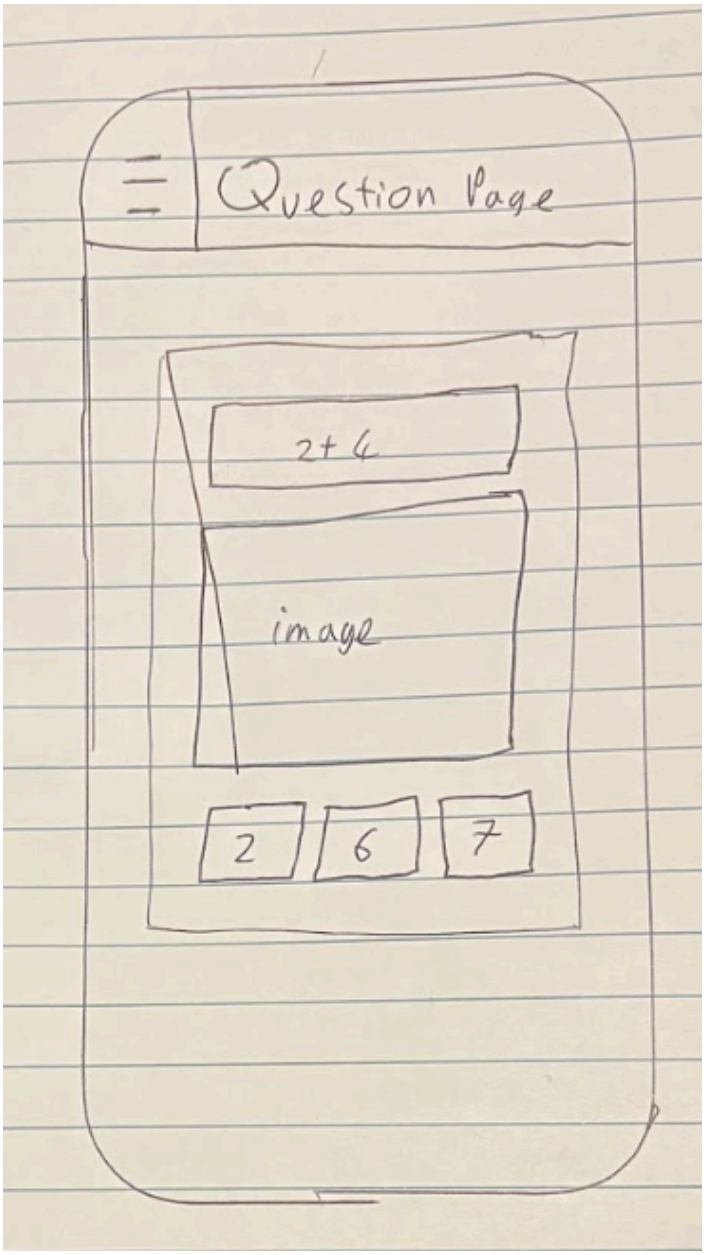


figure 3.4. First design: question page

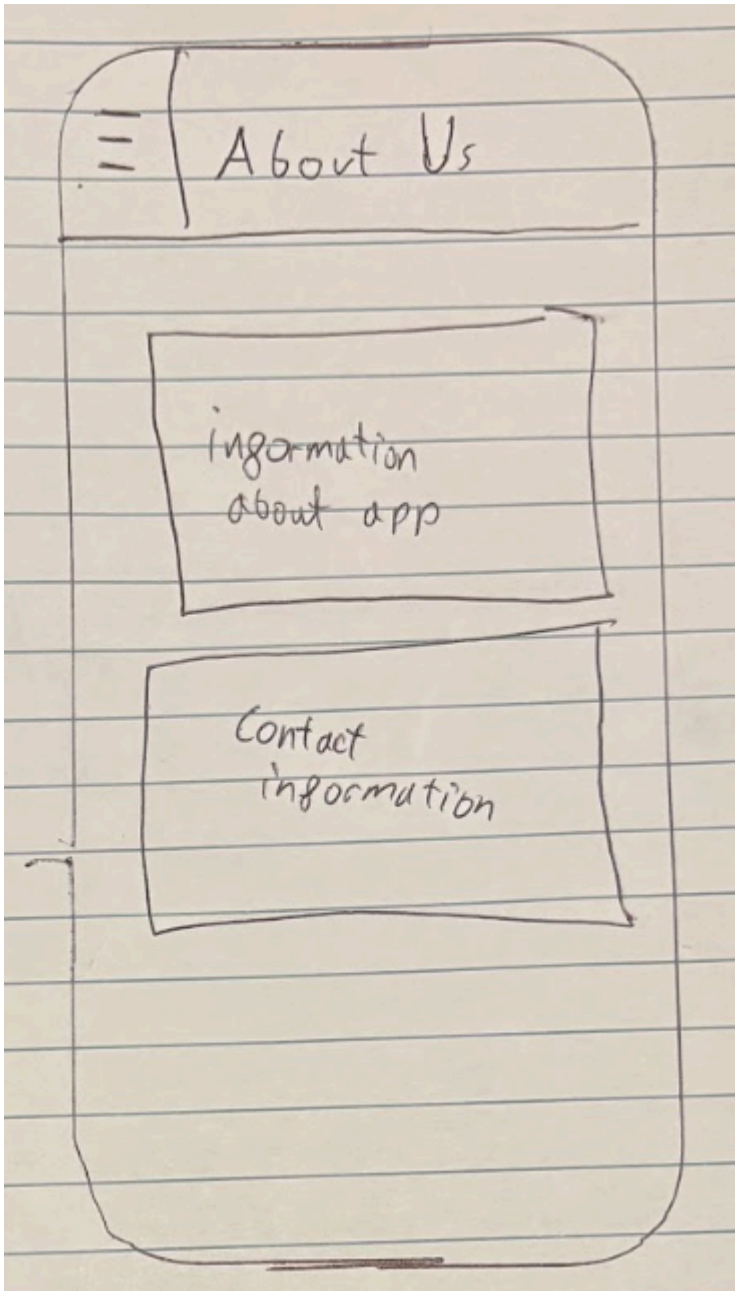


figure 3.5. First design: About Us page

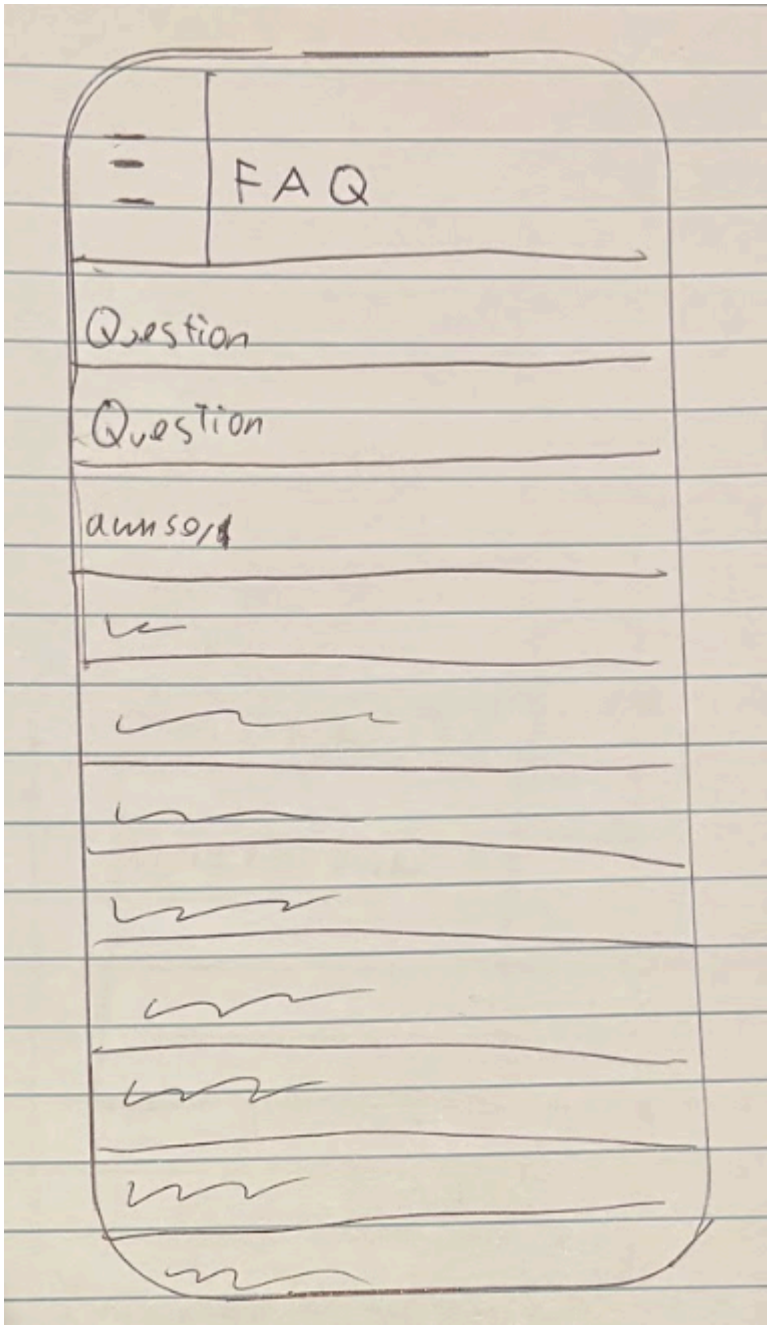


figure 3.6. First design: FAQ page

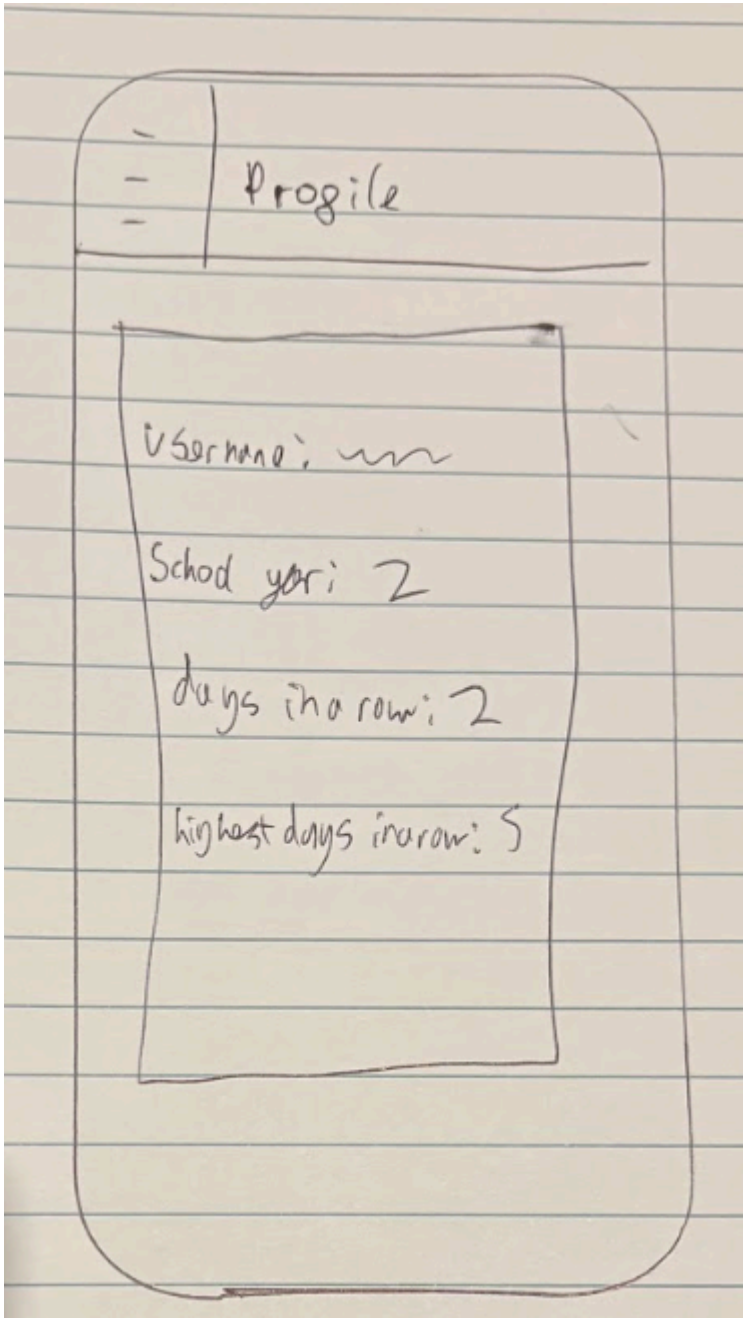


figure 3.7. First design: Profile page

3.5.2. Second designs

Key features were added in this generation of designs such as login, registration, days in a row, score out of 10 on question page and answer feedback as well as colours being added to make the interface more visually appealing..

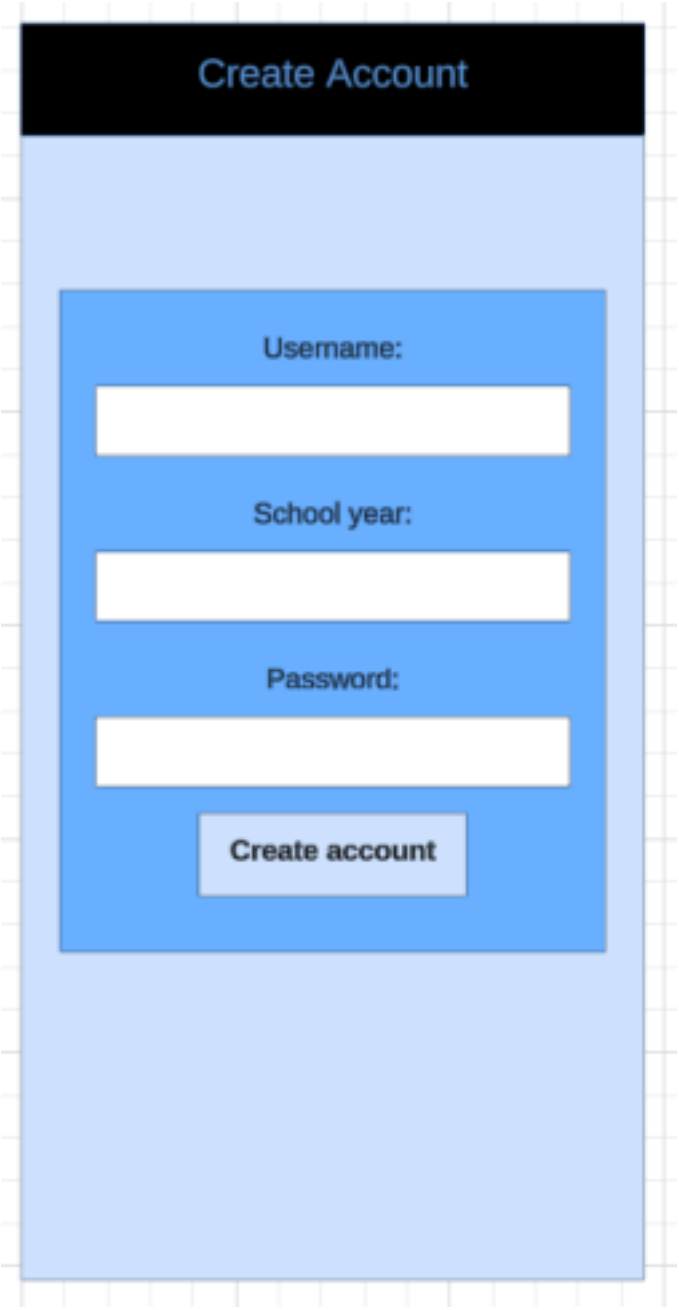


figure 3.8. Second design: Create account

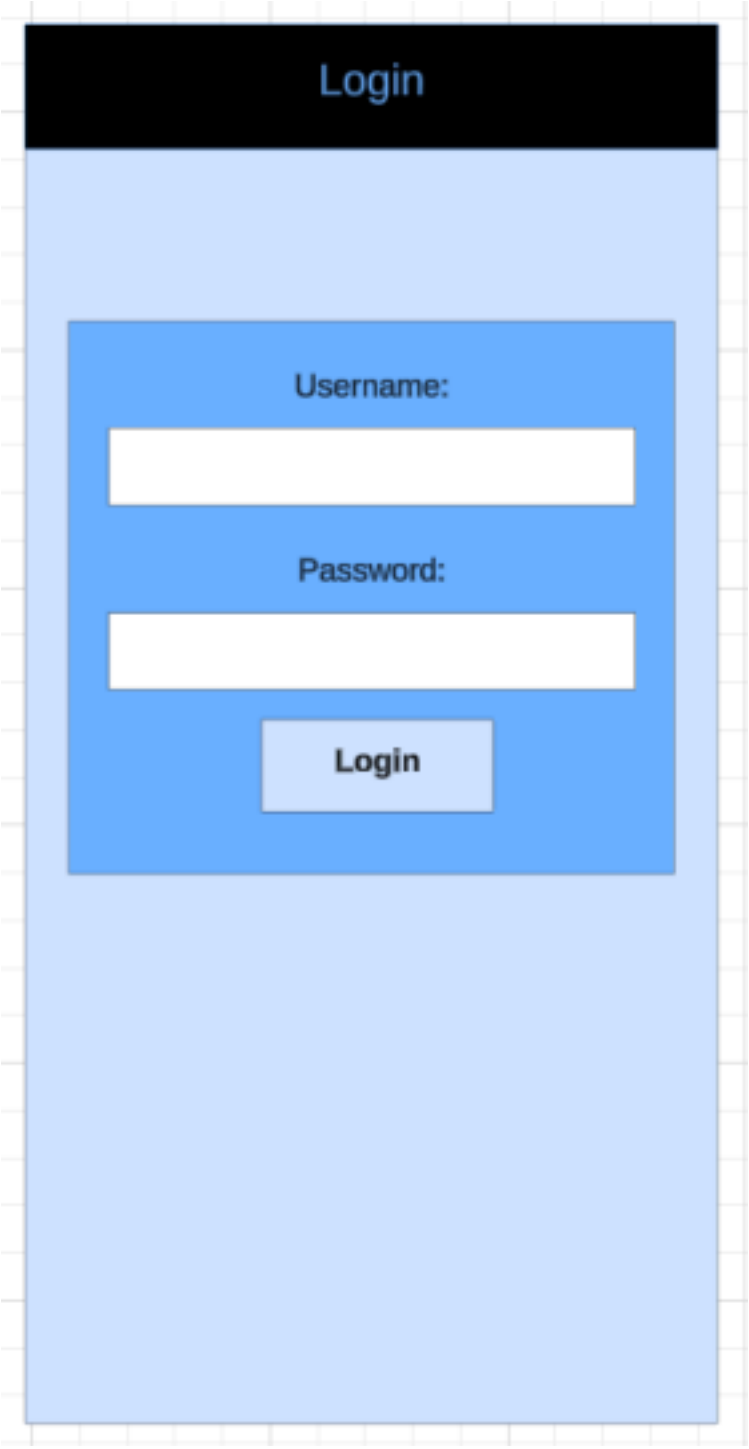


figure 3.9. Second design: Login

figure 3.10. Second design: Question page and menu

figure 3.11. Second design: Answer feedback

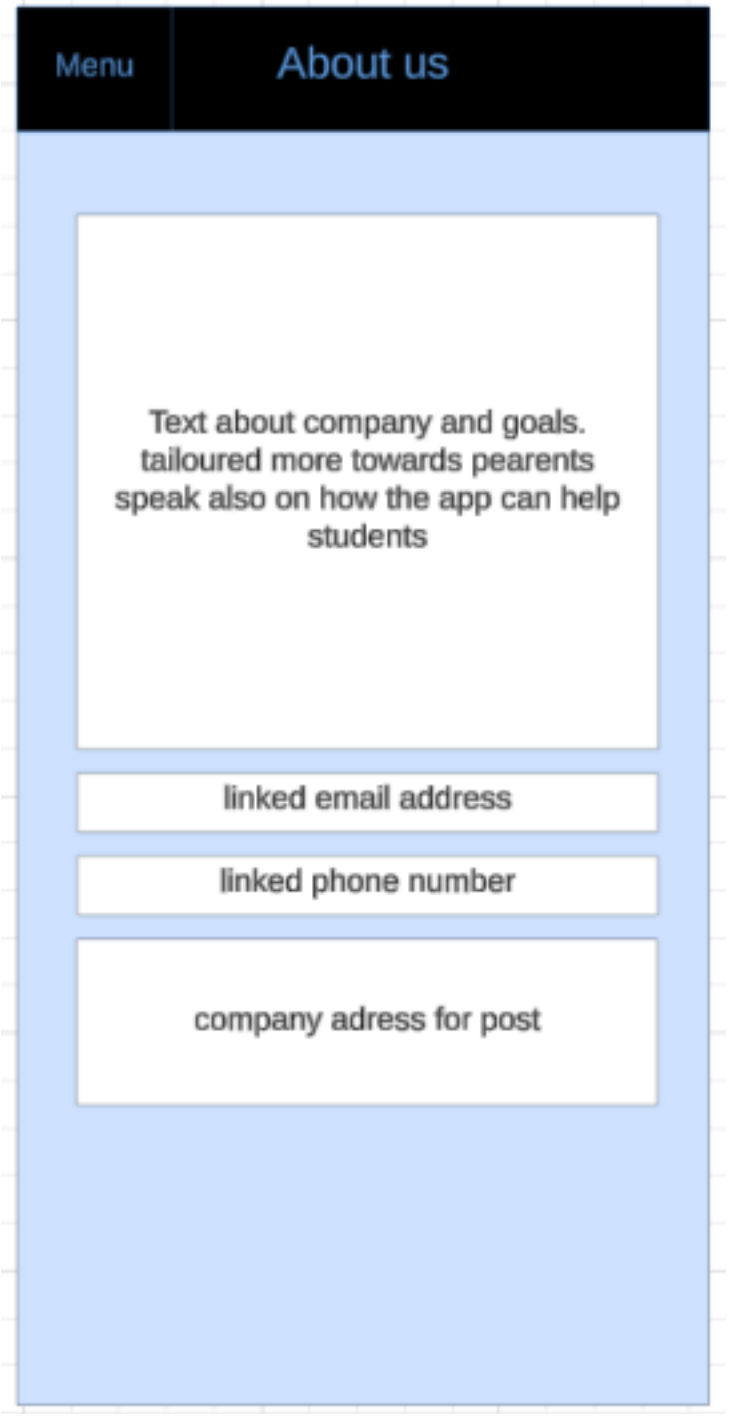


figure 3.12. Second design: About Us page

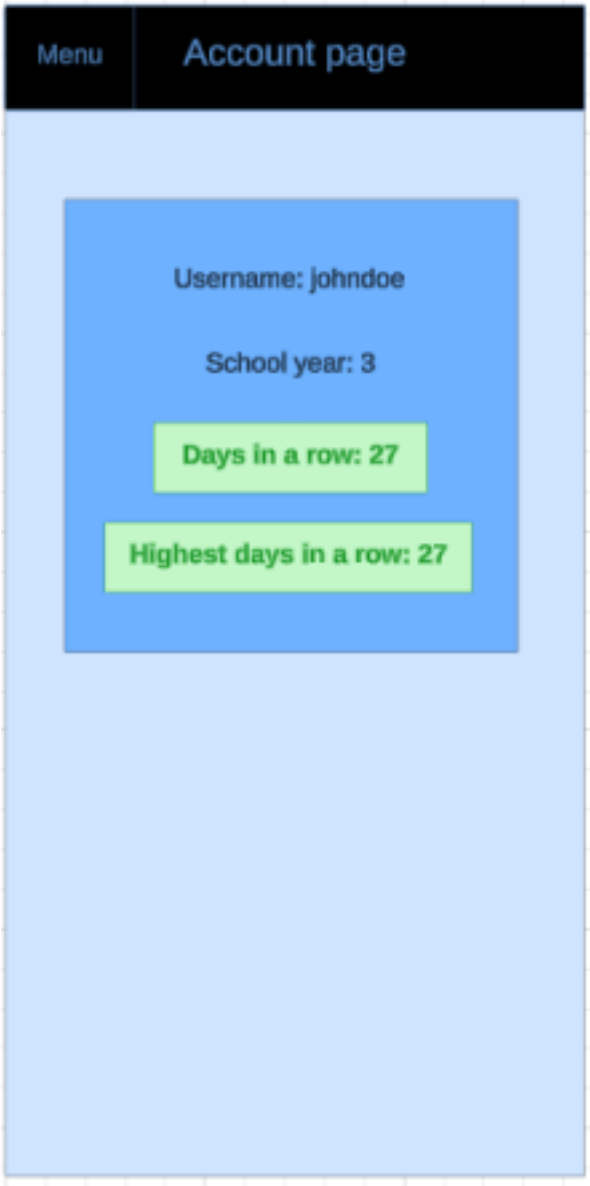


figure 3.13. Second design: Account page

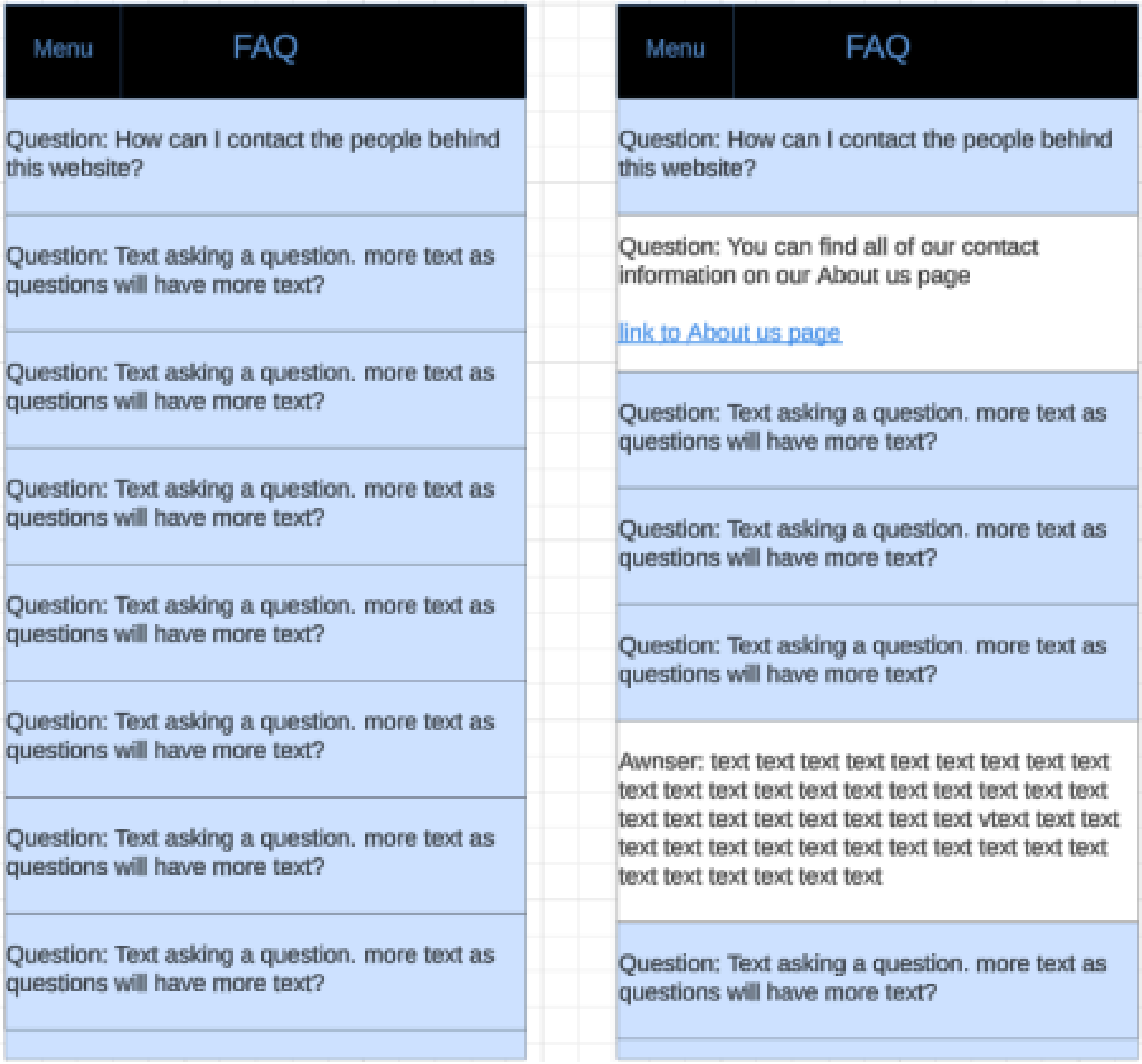


figure 3.14. Second design: FAQ page

3.5.3. Third designs-

Web application instead of mobile due to not all students having a mobile phone especially in early primary school years a web application would be more usable and could be completed in school/on a family computer/on a public computer. This makes the app more inclusive for everyone. Furthermore the literature review supports the overall positivity of maths apps(2.2 Research).

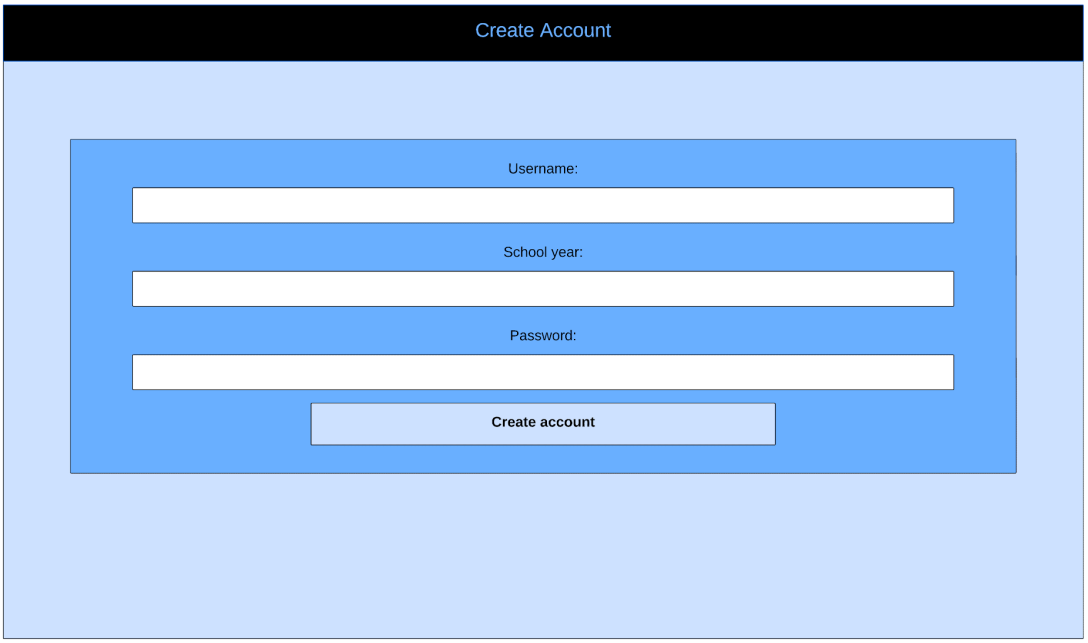


figure 3.15. Third design: Create account

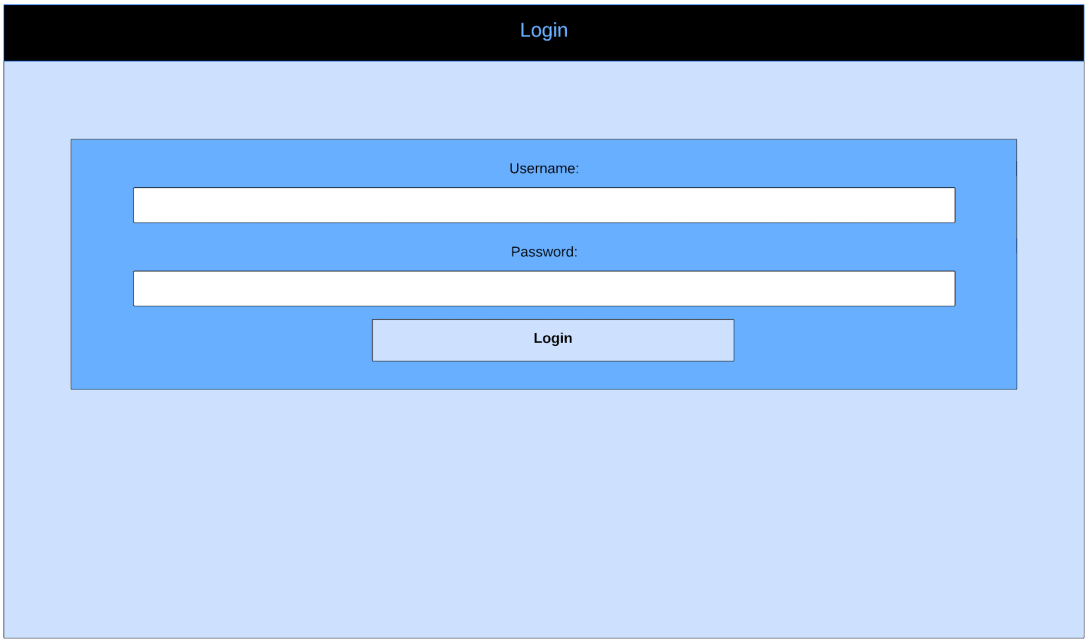
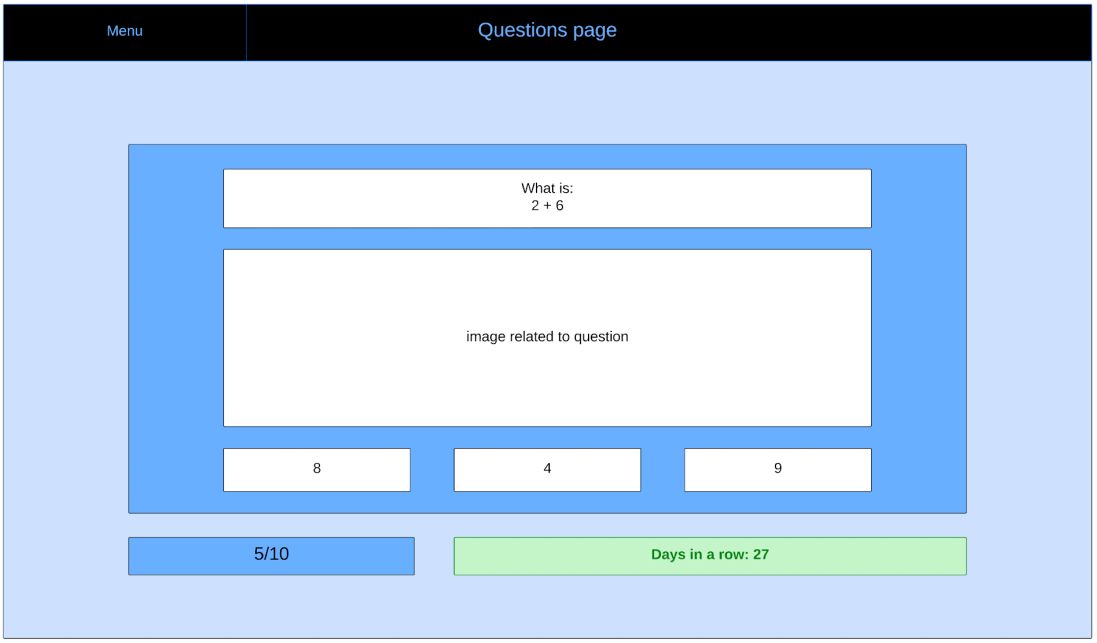


figure 3.16. Third design: Login



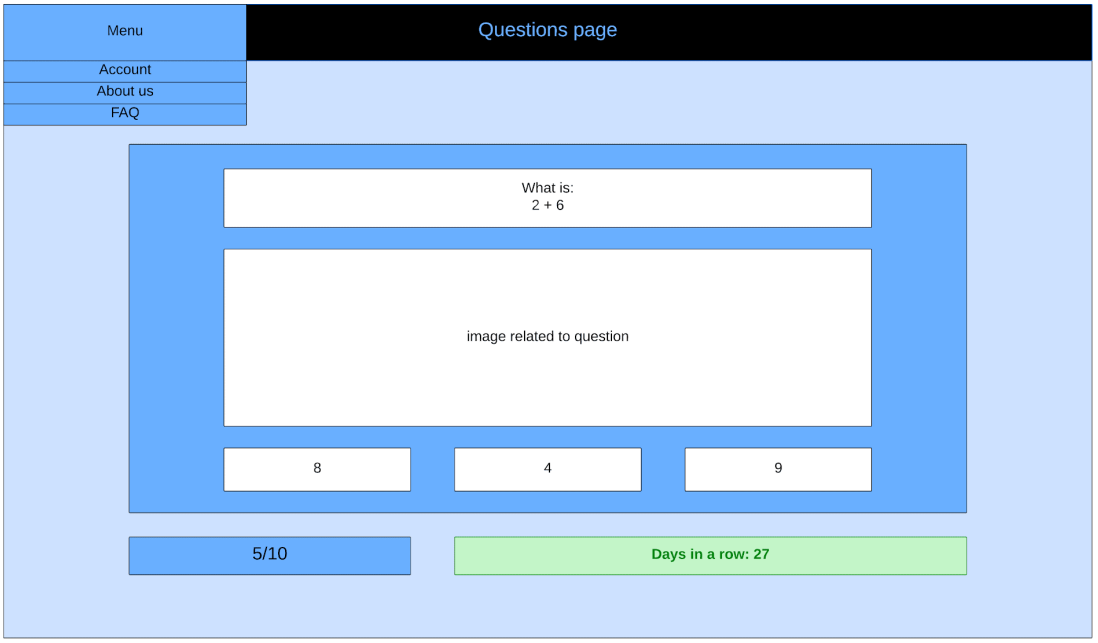
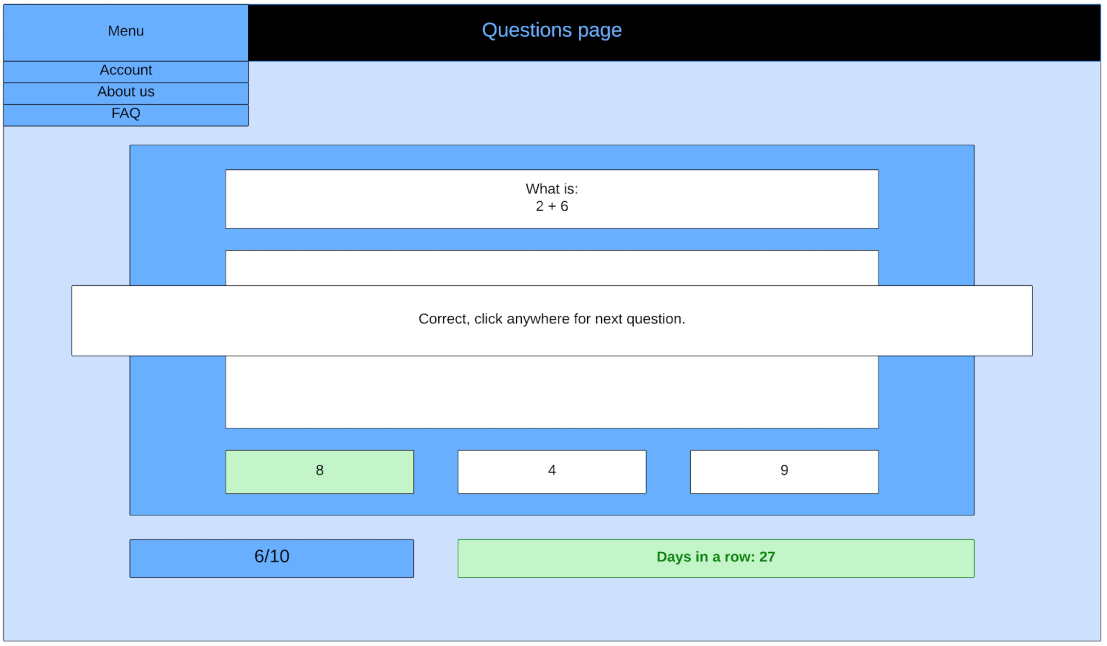


figure 3.17. Third design: Question page and menu



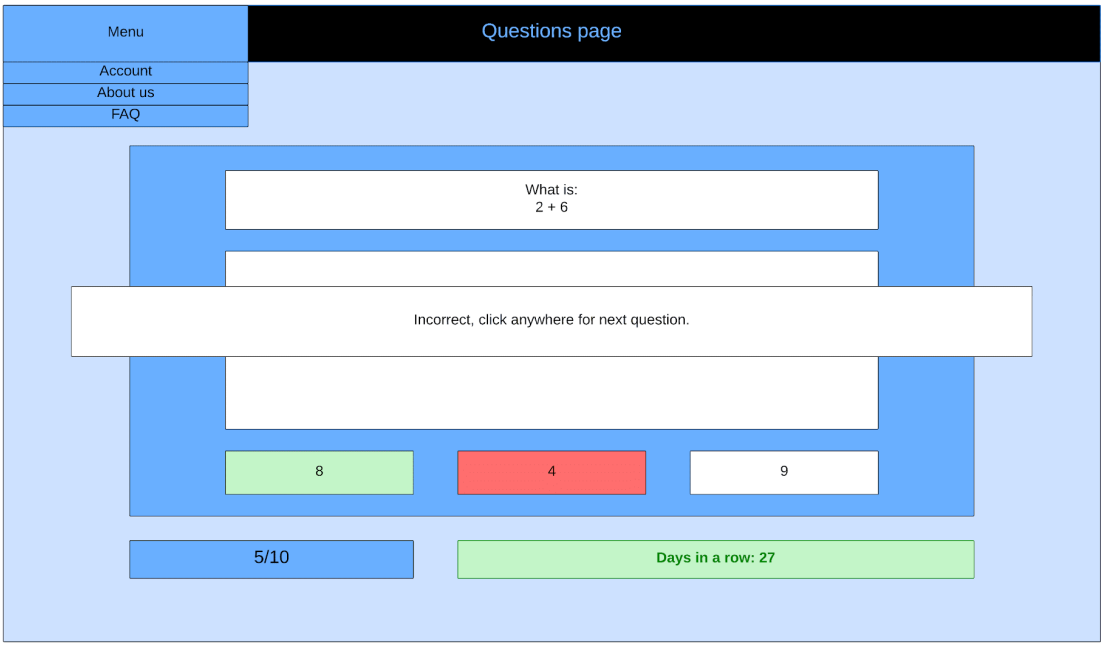


figure 3.18. Third design: Answer feedback

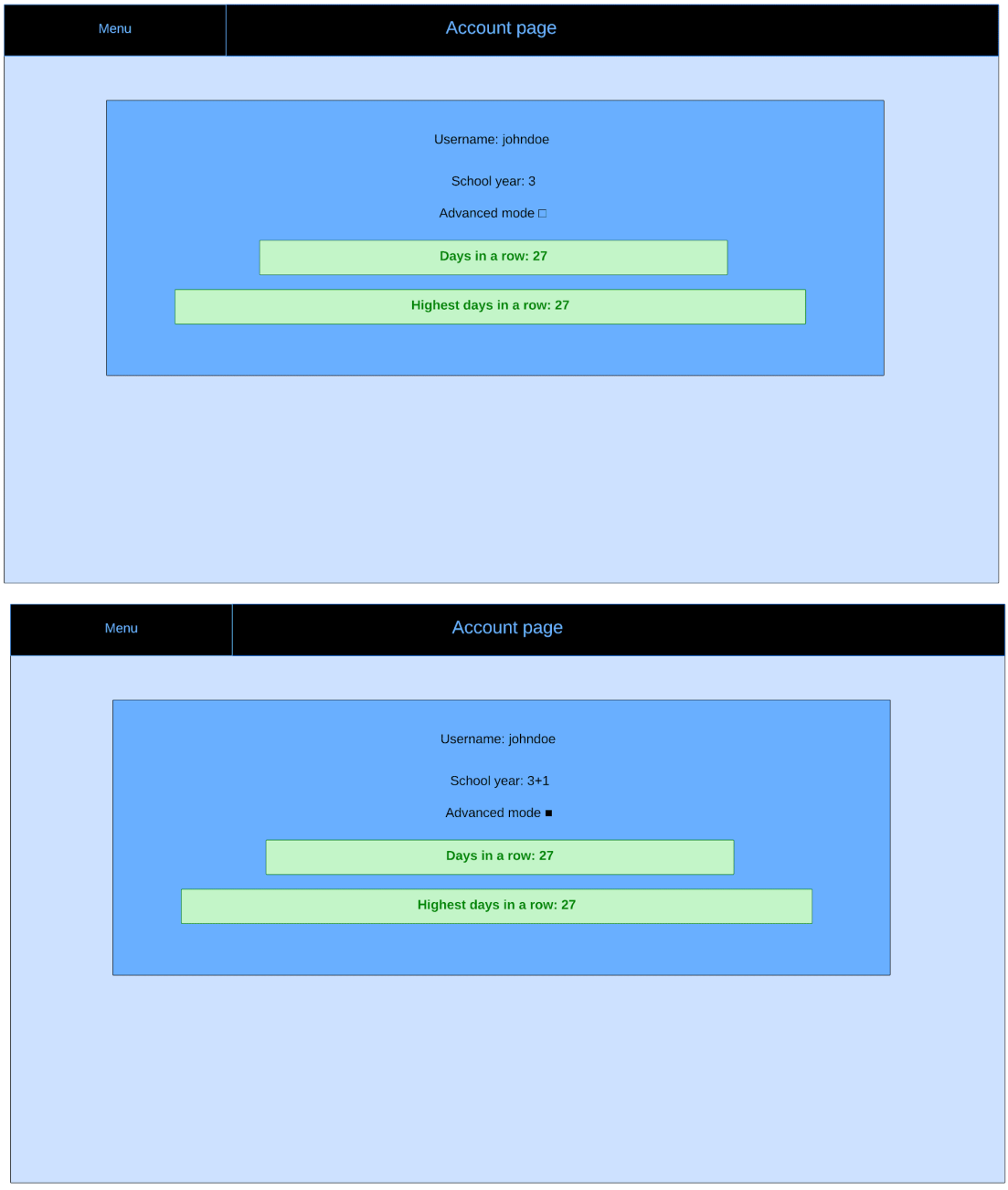


figure 3.19. Third design: Account page

Ability to toggle advanced mode feature was added to the profile page in this generation of designs.

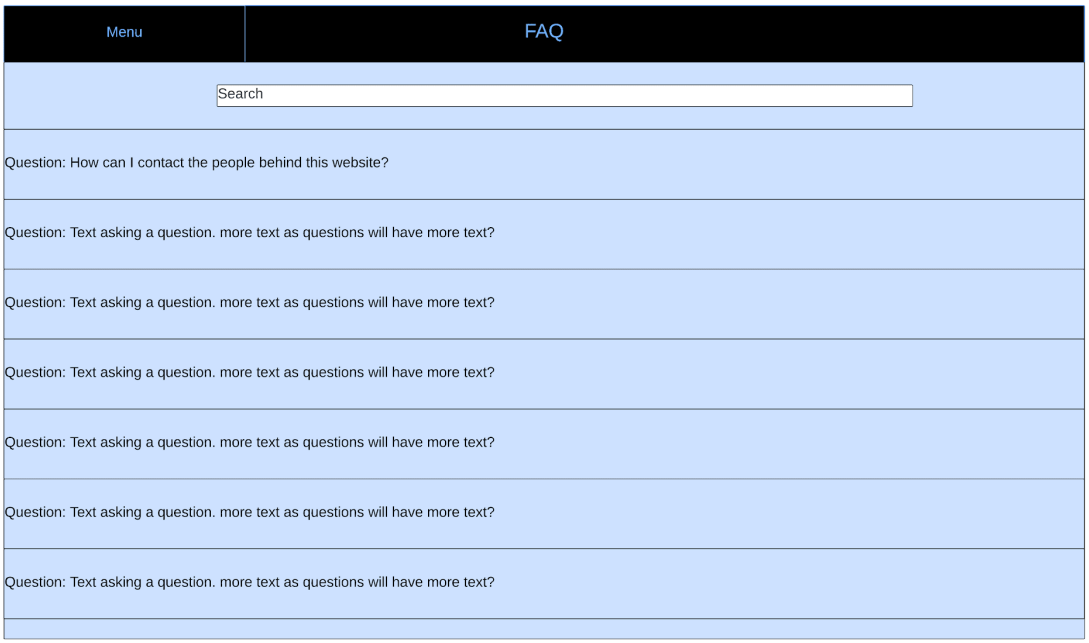


figure 3.20. Third design: FAQ page

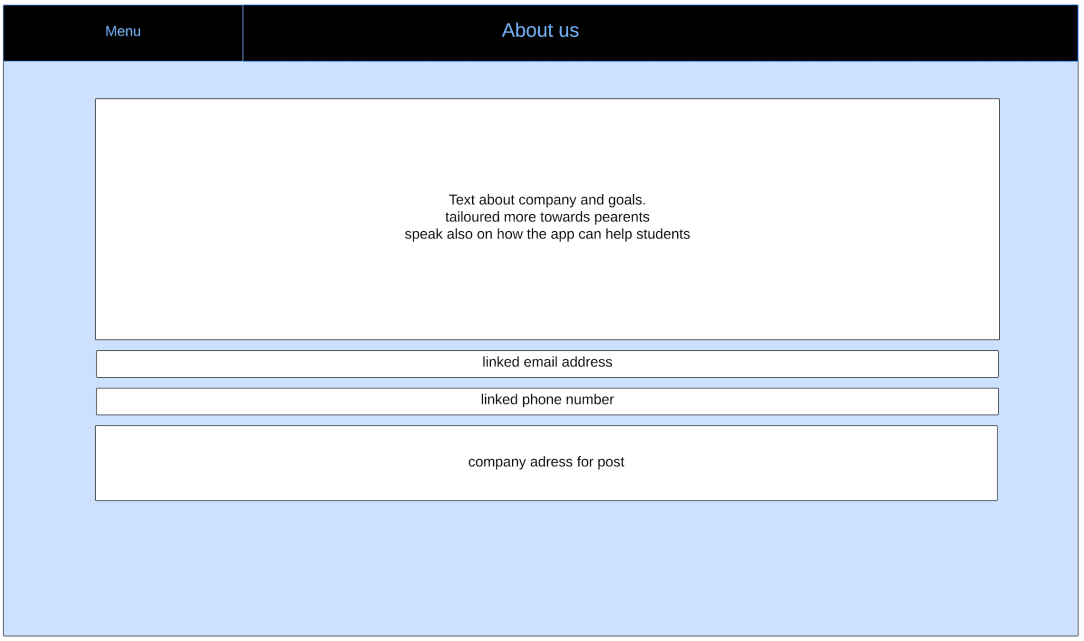


figure 3.21. Third design: About Us page

3.6. Justification of design choices

I designed this app towards my target audience of primary school children. It is very important to design, create and test the app in a safe userable and user-friendly way. With that in mind high usability and being able to gain and sustain child engagement were main goals of the design.

I got significant influence for my designs from my literature review. The research of the other maths apps (2.3. Review of current math apps) inspired ideas such as: a simple interface, immediate visual feedback, changing/random question types, progress tracking and more.

I also took into account Nielsen's heuristics and Schneiderman's Golden Rules as they provide foundational principles for usability and interface design to create an intuitive, efficient, and user-friendly app (2.5. Usability).

Simple interface across all pages- Pages only display necessary information. Options our limited but operation is simple and easy to use even for beginners, partially because there isn't any kind of user error that can't be easily resolved.
clear question and options as the focus of the home page- This is so that as soon as users have logged in they are straight into answering questions, therefore speeding up the overall process of using the app.
Random correct answer out of 3 choices- This feature ensures users will have to read the question and the answers to complete questions, therefore using the app as it was intended.
Images with questions- This is to increase enjoyment/engagement from my target audience.
Clear answer feedback- this is necessary especially considering the age of my target audience.
Track days in a row and score/10 on homepage- This provides feedback to users of their daily goal of questions to get correct as they are answering questions, along with their current streak of

completing that goal. This is so the user has everything they would regularly need in one place providing ease of use.
Simple login and registration that stores only necessary information- This is to make account creation and login quick and easy. I am also being considerate of laws regarding data storage moving forward.
Simple navigation- A clear drop down menu with clear labels for pages allows inexperienced users to navigate relatively easily
About us page- This page will have necessary information and is to inform users about the app, may contain: a description of the app, benefits of the app, why use the app, contact information directed at both children and parents.
Searchable faq- Allows users to find answers to questions about using the app, and parents to questions about the app.
Profile page- Allows users to see their information, days in a row and highest days in a row. This page also allows users to activate/deactivate advanced mode.
Advanced mode- Allows users to easily change between a harder and normal difficulty with visual feedback of if it's on or not.
color scheme- Includes a lot of bright colors to appeal to the young target audience.
Desktop application- Due to not all students having access to a phone for a mobile application.

Chapter 4: Requirements

These requirements took influence from the literature review
These requirements were made alongside the design chapter, at the same time, therefore the designs and requirements influenced each other. This is because I didn't know what I wanted to have as requirements until I started designing the app. Figuring out how I wanted my app to work through designing allowed me to come up with appropriate requirements.

4.1. Functional requirements

Backend database
Firebase database
Stores all data required for users
Read/writes data when needed for app functionality

User authentication control
User must be able to log in
System must take input from the user: username and password.
System must validate the input using firebase
Successfully logging in sends user to homepage
Error messages must be present when the user enters invalid input.
User account creation control
Users can create accounts
Successfully registering sends user to homepage
System must take user input: username, school year and password.
System creates an account if input is valid and stores users input in firebase database.
Usernames must be 4 characters and unique
Passwords must be 8 characters long and include at least 1 number and symbol
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.

User profile
Users can read a selection of their details stored in firebase. These details may be: days in a row, highest days in a row, username and school year.
Users can toggle advanced mode from this page updating their question difficulty and firebase.

Question generation
Users must be able to answer questions.
Difficulty levels must be accurate to corresponding school years.
Randomly generating questions with answers at a difficulty based on the user's school year and advanced mode.
Randomly generating false answers and position of correct answer.
Questions come with the question itself, an image and 3 options as answers.

Answer feedback present after selection.

Daily score
Score will increase with every correct answer until it has reached 10 where it will remain until reset at 12am every day.

Days in a row
Days in a row stores users current streak of completing 10 questions on consecutive days.
Increases by 1 when users achieve 10/10 on any day.
Resets with score if they don’t achieve 10 on any given day.
Highest days in a row would also be stored and updated with days in a row when necessary.

Advanced mode
User has the ability to toggle advanced mode.
toggling advanced mode shows visual feedback
toggling advanced mode adds 1 to users' school year so questions called will be a higher difficulty.
For year 6 users in advanced mode a new difficulty level will be made.

Faq page
Users must be able to find solutions to issues they may have in faq.
Questions must help users understand the app and explain all its features.
Faq page must be searchable.
Option to click on questions to view answers.
Most likely scrollable but depends how many questions.

About us page
Contact information including email, postal address and phone number must be present.
Information about the app must be present.
Email address should have clickable link

Navigation
Users must be able to easily navigate between pages.
App must offer simple navigation.

(these next 3 requirements where added after or during creating designs so may not be included in some or any designs)

Change password
Users must be able to change their password.
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.

Update details
Users must be able to change their username and school year.
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.

Introductory page
Introductory page must be present with a welcome message.

4.2. Non-functional requirements

Usability
The UI should be highly usable across all pages and features.
App should score highly in usability tests.

Security
Password requirements must be met to create an account such as passwords must contain a sign and number.
Users must be able to read/write only their own data.
Database containing user data must be encrypted.
Passwords must be hashed.
Other security measures may be added.

Performance
Features should be highly responsive with little to no wait time.
App should be able to hold 100 users as this is the capacity of the free plan on firebase.
Question generation should take less than a second per question with stable internet connection.

Age appropriate difficulty
Question difficulty for each school year should be researched and accurate to the uk national curriculum.

Scalability
App should be able to cope with an increasing amount of active users.
App should be able to have updates where features or more questions are added.

Reliability
Features should function as intended everytime they are used.
There should be no long loading screens

Availability

App must ensure a minimum 99% uptime to ensure users have access to it.

Fault tolerance
App should be highly tolerant to user error without the ability of causing lasting damage to your account.

Compliance
App must comply with all relevant data protection laws such as DPA 2018.
App must comply with all university rules and regulations.

Compatibility/accessibility
App must be compatible across different browsers and operating systems on pc.

Quality assurance
Multiple testing procedures will be performed on the system to ensure everything is working as expected. This could include: usability testing, functionality testing, security testing, compatibility testing and difficulty testing.

Chapter 5: Development

5.1. Introduction

I implemented Math Club using Flutter as my primary language on Visual Studio Code, with Firebase integration providing backend services for data storage and user authentication. This chapter documents my development journey, explains features with screenshots, talks about problems encountered as well as justifying changes from the design chapter.

5.2. Stages of development

Navigation- I started by implementing blank pages as placeholders and adding a uniform banner to all pages. The home/question page has a dropdown menu to all pages, other pages have a back button. This allows navigation between pages and helps me view pages that I'm implementing.
About us- Then I implemented the layout for the about us page. Other than an email link, all i did was add containers with mostly place filler information
Faq- I had previously created a searchable FaQ for a project so I used that as a starting point for this one. I removed some features that I didn't want, put placeholder questions and answers and made some design changes to better suit my project and its themes. This page allows you to view questions that are clickable to view answers and is also scrollable
Login/Registration- Next I just created the interfaces for login and registration which were both basically forms with a button to finish and get access to the app(when firebase has been implemented). There is a link between these pages.
Information stored for registration: email, username, school year, password
Information required for login: email, password
Firebase- This stage proved to be difficult, but I set up the firebase database to store the user data required. I had to initialise the database on the firebase website and write it into my code. I have a lot of algorithms for firebase defined in auth.dart, these are for a variety of functions such as updating values from the database or logging in to the app.
I also created a database file that defines the structure of the database and the values it will store.
Profile Page-This page required retrieval of users information from firebase with this information being displayed. It also uses functions in auth.dart.
Information displayed: username, email, school year, advanced mode, days in a row, highest days in a row
Change details- This feature wasn’t in design but seemed like a necessary feature to add. You can now open two forms. One form changes your password by requesting the old password and new password twice. The other asks you for your new username and school year.
Advanced mode- Advanced mode can also be activated here, it is another piece of data stored on firebase and is used in calling the questions. If it is active advanced mode is 1 if it is not active advanced mode is 0, school year + advanced mode = learnage which is a value passed from question page to questiongen and is used in determining which set of generated questions a user will have to answer questions from.
Question page- This is the most important page in my app. Based on the user's school year and advanced mode an algorithm calls a question using questiongen.dart. This algorithm returns a question, answer, and two false answers.
Random questions and answers- questiongen generates random questions using learnage. A question type is determined by a random number being generated between 0-1 with each question type having an equal probability range. Each type of question (x,+,-,/,written) will have different ranges for its first and second numbers, these ranges increase if learnage is higher and the numbers are random within the ranges. There is no lower range for these numbers so difficulty varies. The question

generation process may have other conditions like the 1st number must be larger than the 2nd for subtraction, and division is calculated differently using the answer*the divisor to ensure a whole number answers. Question is displayed on the question page and the correct answer is one of 3 buttons at random with the false answers in the other 2 buttons. The correct answer will always be labelled with the correct answer to the question. The algorithm is long however the learnages are repetitive so this made it easier.

Images- A random image will be selected out of a folder in assets with every question. It is selected by generating a random number between 1 and number of pictures because each image file is named with a number. Images were created using chatgpt 4, I described to the AI very specific prompts of ideas I had for pictures. There are 56 images in total (see Appendix C: Images used in Math Club)

Score/10- A number that resets to 0 at 12am everyday. For it to increase days in a row it must reach 10. It will increment by 1 upon every correct answer up to 10 where it will remain until reset.

Days in a row- every time Score reaches 10/10 days in a row will increment by 1, if this is the new highest days in a row highest days in a row will change to current value for days in a row. If the score does not reach 10 when reset occurs or if a reset has been missed days in a row will be set to 0.

Error messages- I implemented a detailed system for error messages handling for registration and change details or password. System has multiple catches and on error clearly explains issues by saying what field has an issue and what that issue is, this allows users to fix that issue easier as they know why they can't register or change their details.

Introduction page- I added a basic page to welcome users and show application name.

About us text- To write this text I tried to explain the app and its benefits in a meaningful manner. For this page I also considered parents, teachers and children as the target audience. I pasted this text over filler values.

Faq text- To write this text I tried to think of questions users may have and answered them as clearly as I could. For this page I also considered parents, teachers and children as the target audience. I pasted this text over filler values.

Research- Research was based on the national curriculum. I based my upper ranges for numbers involved in questions on this. I then replaced filler values in code with researched values. This is explained in more detail under the research heading.

Written questions- Worded questions and answers are written manually. There is a random chance that this will be the selected question type (20%). In these cases each difficulty level has 25+ possible questions that could be randomly selected. This means the chance of any worded question being selected will never be higher than 0.8% (4% of 20%) reducing the likelihood of repetition. These questions allow me to experiment with different styles of questions that would be difficult to generate. Questions can be unique from each other and add variety to the experience. These questions do however have limitations such as they are time consuming to write, there is risk of user error, and they have less possibility as they have set answers and figures that will not change. Numbers and subjects in these questions were based on the national curriculum like ranges for generated questions. More questions could easily be added to each difficulty. For users in year 6 using advanced mode they will have the normal year 6 questions as well as 15 more difficult questions. These questions are supposed to vary in difficulty within their respective learnages. There are 165 drafted worded questions in appendices however these vary slightly to the worded questions in code as changes were made but the table wasn't updated. (see Appendix D: Worded questions)

5.3. Problems encountered

About us covering 2 lines on navigation- I couldn't resize the box the text ‘about us’ was in. I recall this problem taking a long time to fix when it seems trivial. For that reason this problem is unresolved and may well stay unresolved.

Can't resize menu- Similar to about us issue, this problem was considered insignificant. Menu feature is fully functional and in my opinion usable, so I stopped trying to change it. This problem is unresolved and i dont see any reason to try and resolve it.

Login with username and email- login was initially meant to be with username and password, however I decided to change to email and password for login. I wanted to add a secondary option to login with a username but I couldn't get it to work. This problem is unresolved but may be resolved in the future. Update- this has been resolved by resizing the menu.

Couldn't link firebase to application- I had problems implementing firebase. I thought i had done everything correctly however it wasn’t working. After checking my code multiple times I found a small issue that stopped firebase from operating. The issue was to do with incorrect code to support firebase. It was resolved early in development.

Not loading profile page- users couldn’t open their profile page. This is because it calls data from firebase and users didn't have read access. I changed this in the rules on firebase, this resolved the issue.

Email verification- Early in development I tried to create an account verification feature. I created a page and got emails to send, however the app wasn’t tracking verification and just let all users in when they pressed i've verified. This feature isn’t vital for the app so the verification page and features aren't in use. Issues have not been resolved but may be resolved in the future if I decide to implement parental consent.

Separate error messages for login- I attempted to have two error messages, one for incorrect password, and one for email not being registered. I couldn’t get this to function and decided it isn’t a big issue and having one error message is ok (5.5. Features explained with screenshots). On the other hand this improves security as users have less information as to why they can’t access any given account. It hasn’t been resolved but may be resolved in the future.

Users age automatically updates- I couldn’t get this feature to function properly, however I added a feature where users can change their school year on their profile page. Problem hasn’t been directly resolved but I have used an alternate solution that gives users freedom to change their details when they want to. I may implement this in the future but it isn’t important now users can manually change their details.

Can’t call questions- I was having issues with generating questions based on school year and advanced mode. I was trying to call them directly to questiongen but questions would not generate. It only worked when I called the two values in the question page and combined them using addition to get learnage. Question page would then pass learnage as input to questiongen to generate questions and answers with correct difficulty. This problem was resolved.

Days in a row resetting- Days in a row was resetting for incorrect answers when the user did not have a 10/10 score on that day. There was a line of code in on answer causing this. This issue was resolved when I deleted the line of code.

Days in a row resetting- Days in a row would reset if score wasn’t 10/10, however if score is 10 even if reset hasn’t happened for 2 or more days, days in a row wouldn't get reset. I added a second condition for resetting days in a row that checks if there was a reset the previous day, if there wasn’t then days in a row is reset. This issue is resolved

Back on login- back on login was taking users to the question page. To resolve this I deleted the back button and if users go back using browser functions they would be redirected to introduction page

Back on homepage- back on homepage was logging users out, I didn’t want this to be an option of logging out so I disabled the back button from changing page.

Change password- There was no feedback for password updating or wrong password and when I checked password wasn’t changing. To fix this I made new functions to handle changing passwords.

5.4. Research

To get the values I would use for each school year I studied the national curriculum by the department of education (DfE, 2013). Below is text I found on the national curriculum along with the values I chose based on said text. After I had these values I just had to replace the filler values in my code and make a small change to the algorithm as year 1 doesn't actually cover division or multiplication. Each pair of values represents upper limits for 2 numbers in a question. For multiplication, addition and subtraction these values represent the operands, however for division one is the divisor and the other is the answer and they are multiplied to get the first operand. Question difficulty should vary within the curriculum so questions offer a challenge but do not dishearten users who are struggling. See (Appendix E: Text directly from national curriculum with chosen values) for research with chosen values

5.5. Features explained with screenshots

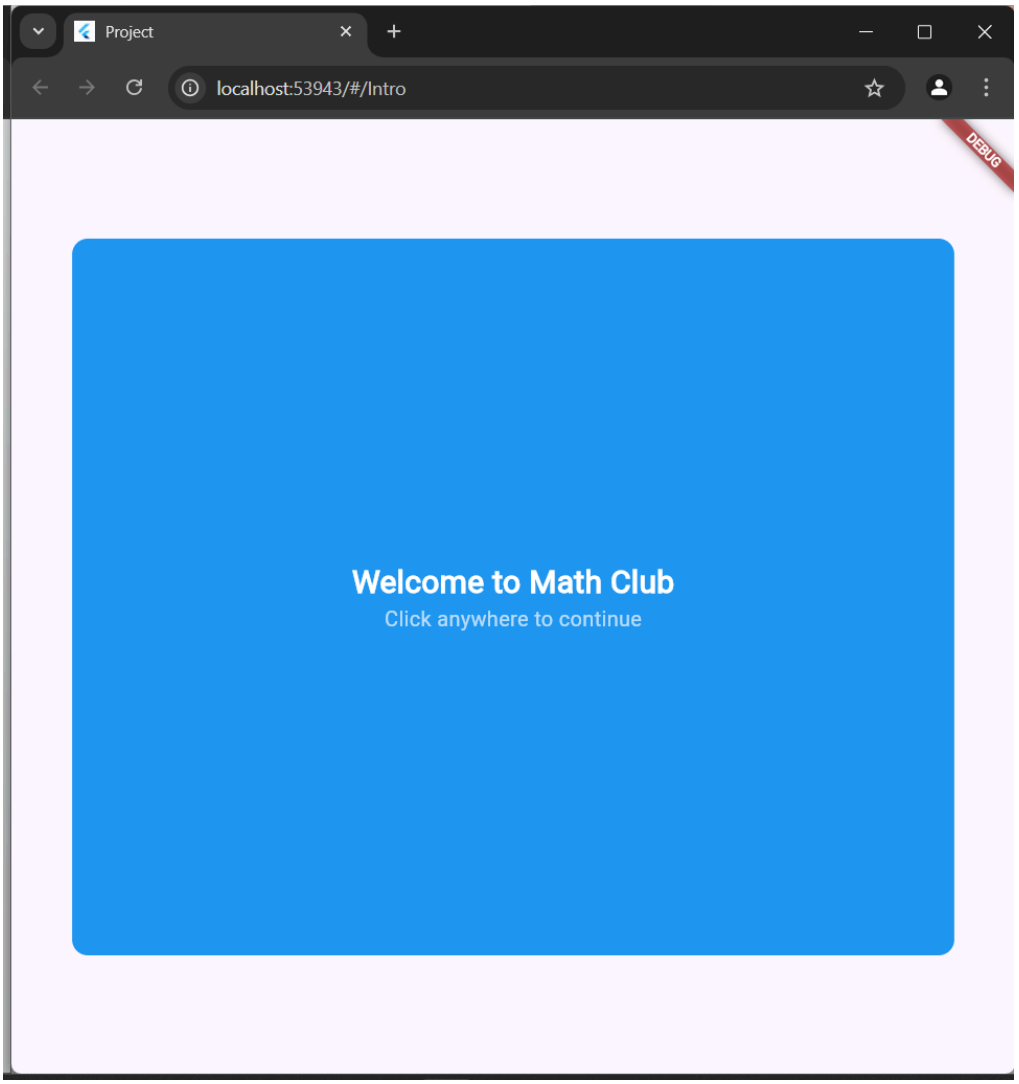


figure 5.1.Screenshot of implemented introduction page

Introduction page to welcome users and display app name. Clicking on the blue square sends users to login.

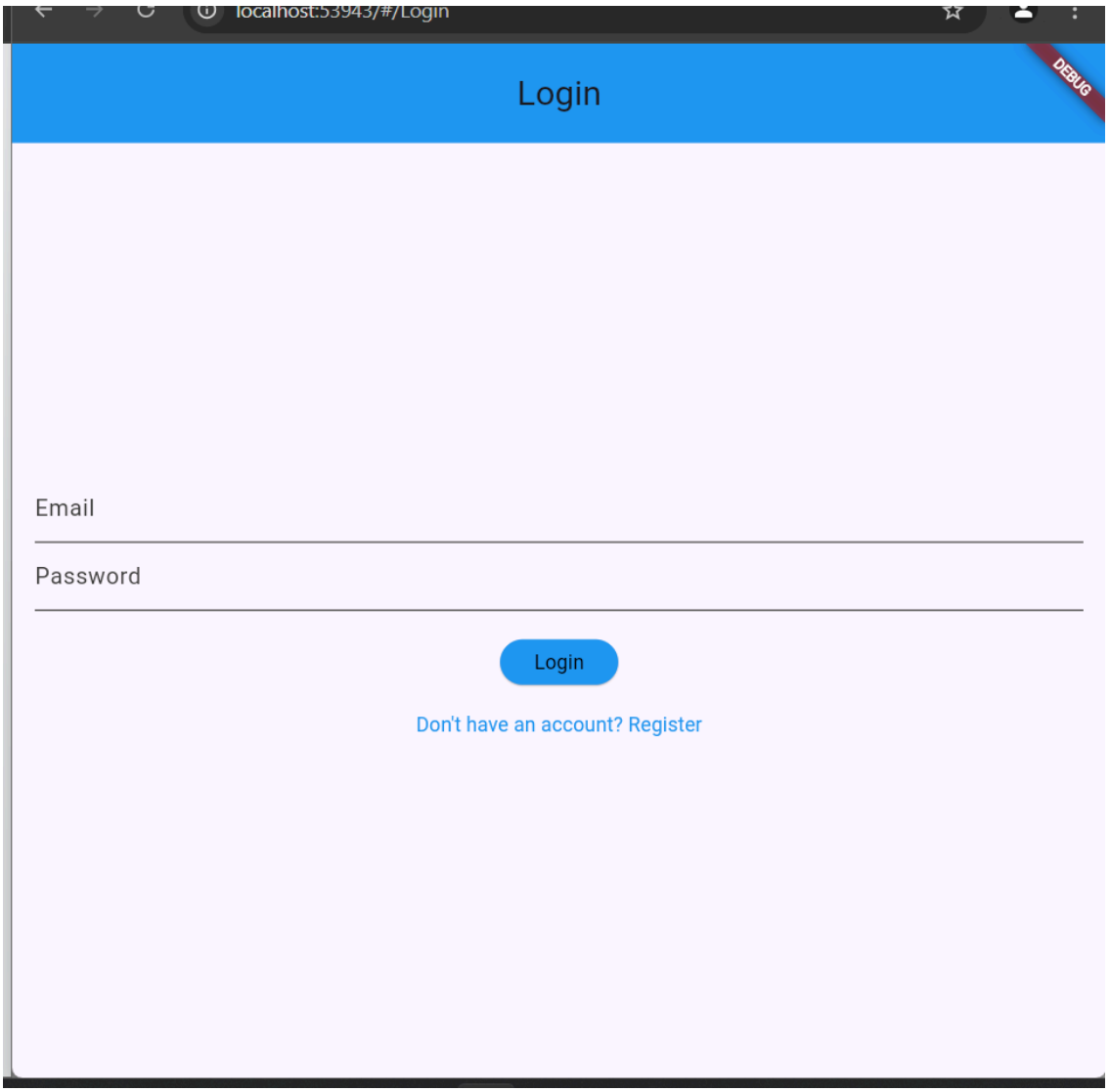


figure 5.2.Screenshot of implemented login page

Login page so users can access accounts and data stored to their accounts.

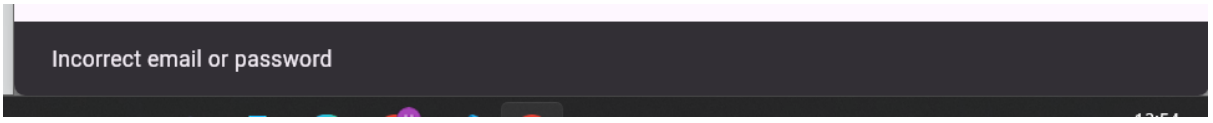


figure 5.3. Screenshot of login page error message

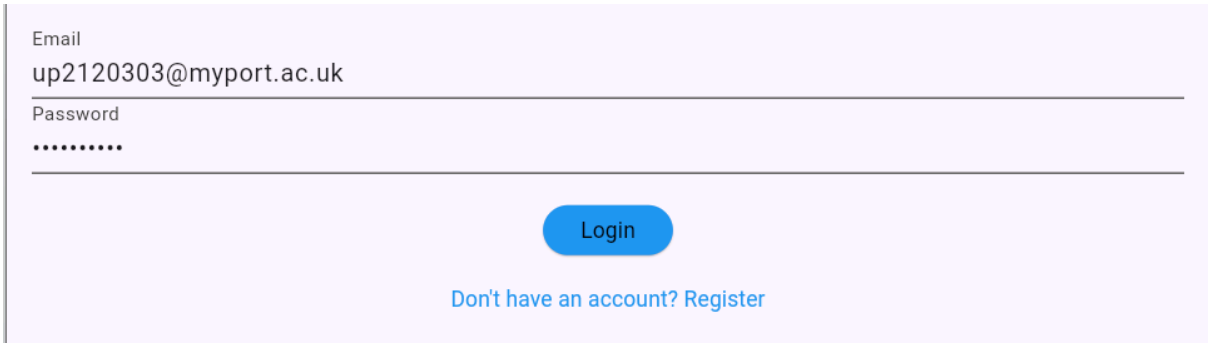


figure 5.4. Screenshot of text entered into login fields

Functional login, if the user enters correct details and presses login their data is called from firebase and they are directed to the home screen.

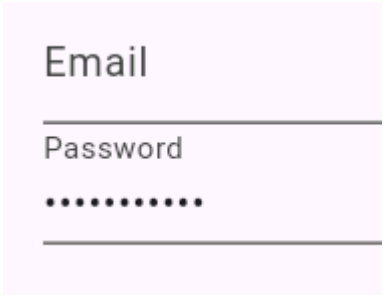


figure 5.5. Screenshot of obscured password text in login fields

Password text obscured

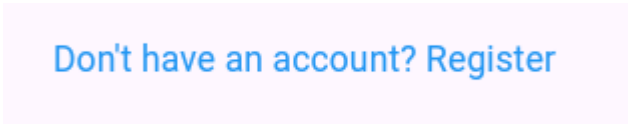


figure 5.6. Screenshot of link to register page button

Link to register for users without accounts.

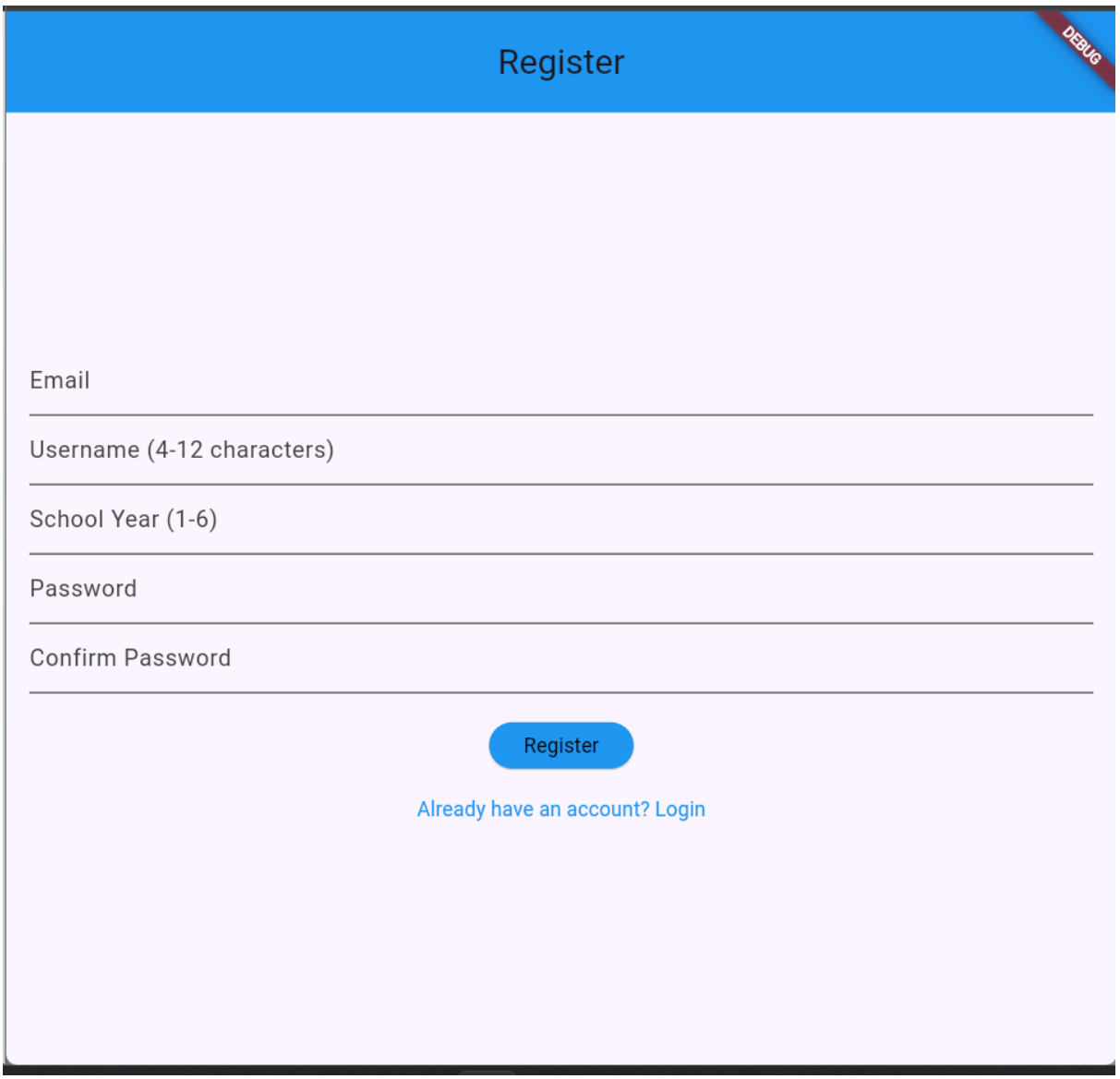


figure 5.7. Screenshot of implemented register page

Register page so users can create accounts.

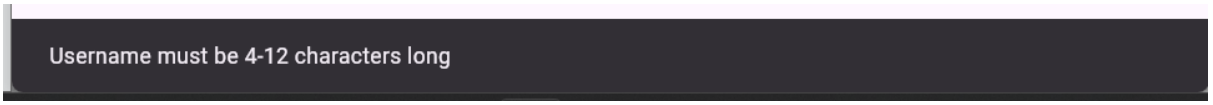


figure 5.8. Screenshot of username error message

figure 5.9. Screenshot of school year error message

figure 5.10. Screenshot of email error message

figure 5.11. Screenshot of confirm password error message

figure 5.12. Screenshot of second username error message

figure 5.13. Screenshot of second email error message

Multiple error messages to clearly inform users why they can't register, as there are a lot of fields with limitations to what users can use. This helps usability and makes account creation easier.

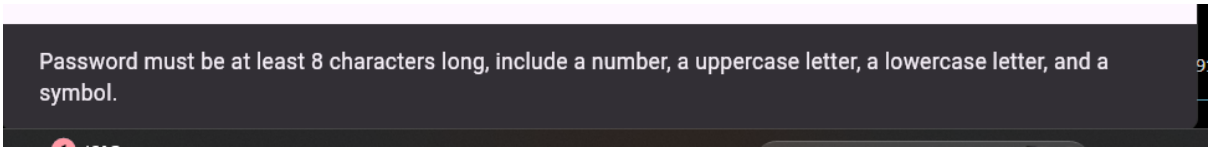


figure 5.14.Screenshot of password error message

This error message is displayed for a duration of 6 seconds as it is longer than the other error messages.

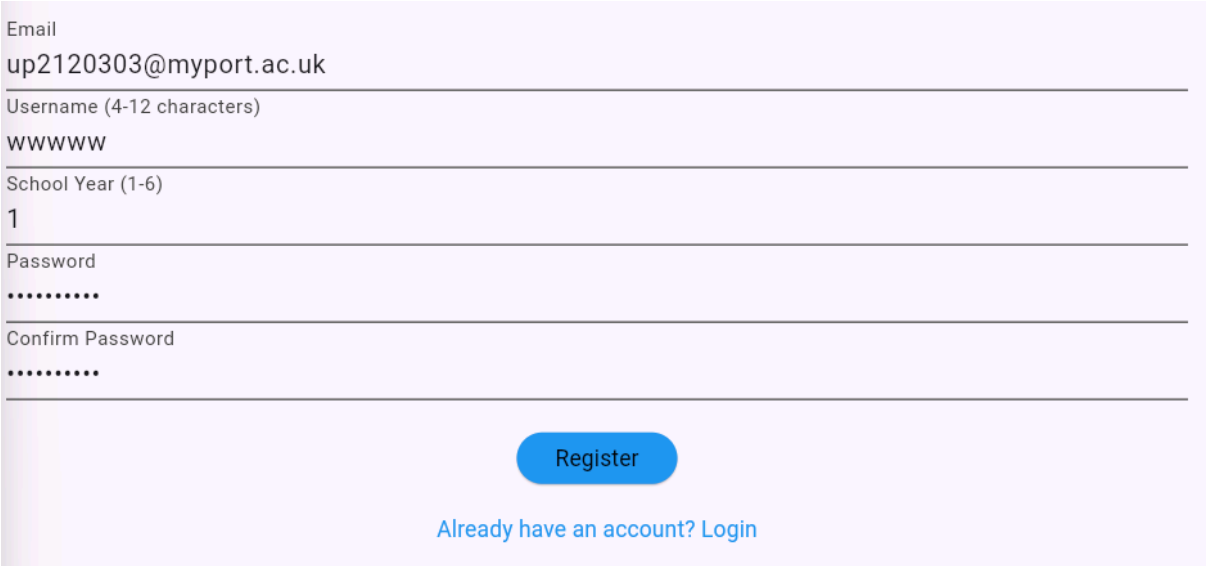


figure 5.14. Screenshot of password error message

Functional register, if a user enters valid details and presses register their data is saved to firebase and they are directed to the home screen.

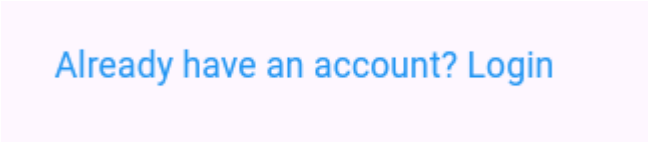


figure 5.15. Screenshot of text entered into register fields

Link to login for users with accounts.

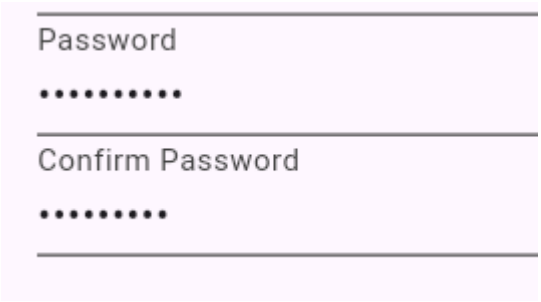


figure 5.16. Screenshot of link to login page button

Password text obscured

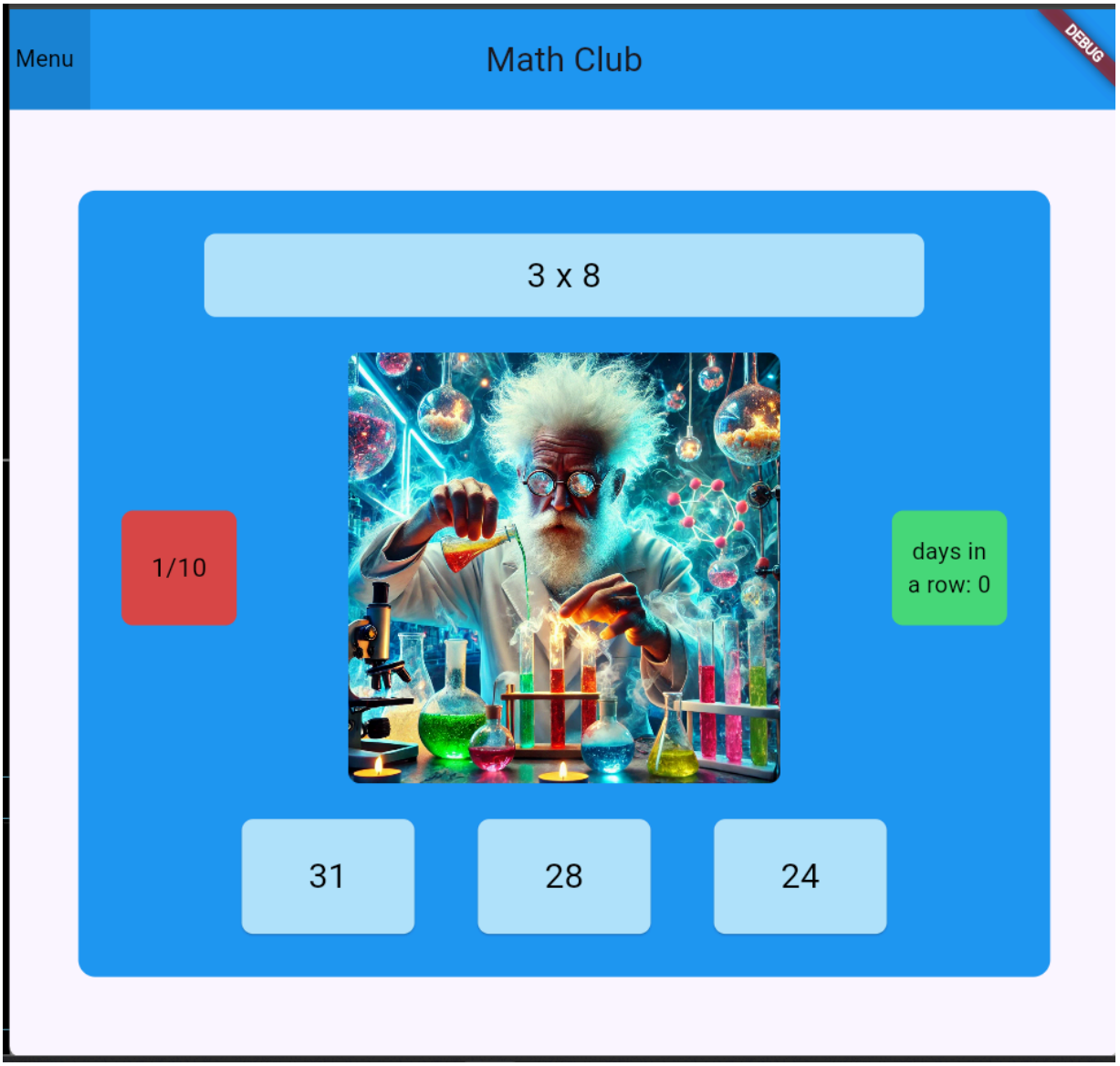


figure 5.18. Screenshot of implemented question page

Home/question page is the question page and the focus of the app. I have kept the interface simple with only necessary or helpful features present for usability reasons.

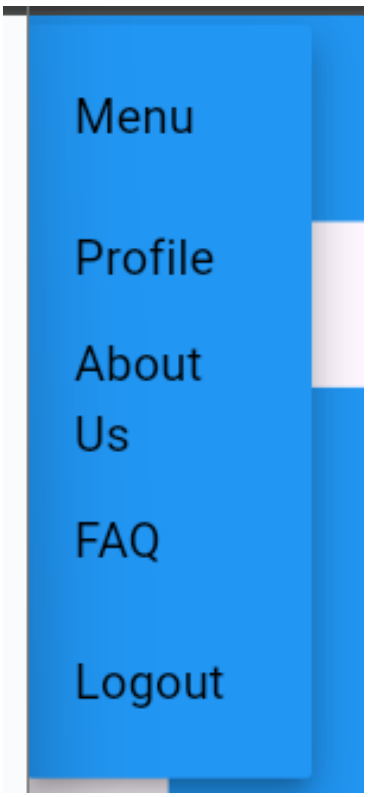


figure 5.19. Screenshot of dropdown menu

Drop down menu unique to homepage/math club. The Logout button logs users out and other buttons direct users to pages. Pressing the menu again will close the menu.

figure 5.20. Screenshot of question

Random questions are generated by randomly selecting a type of question for example multiplication, Then using 2 separate upper limits for the 1st and 2nd numbers being multiplied. These upper limits change depending on question type and learn age. Other factors include:
-Division questions are made by generating the answer and the divisor. Answer x divisor = 1st operand.
-learnage 1 only has addition, subtraction and worded questions.

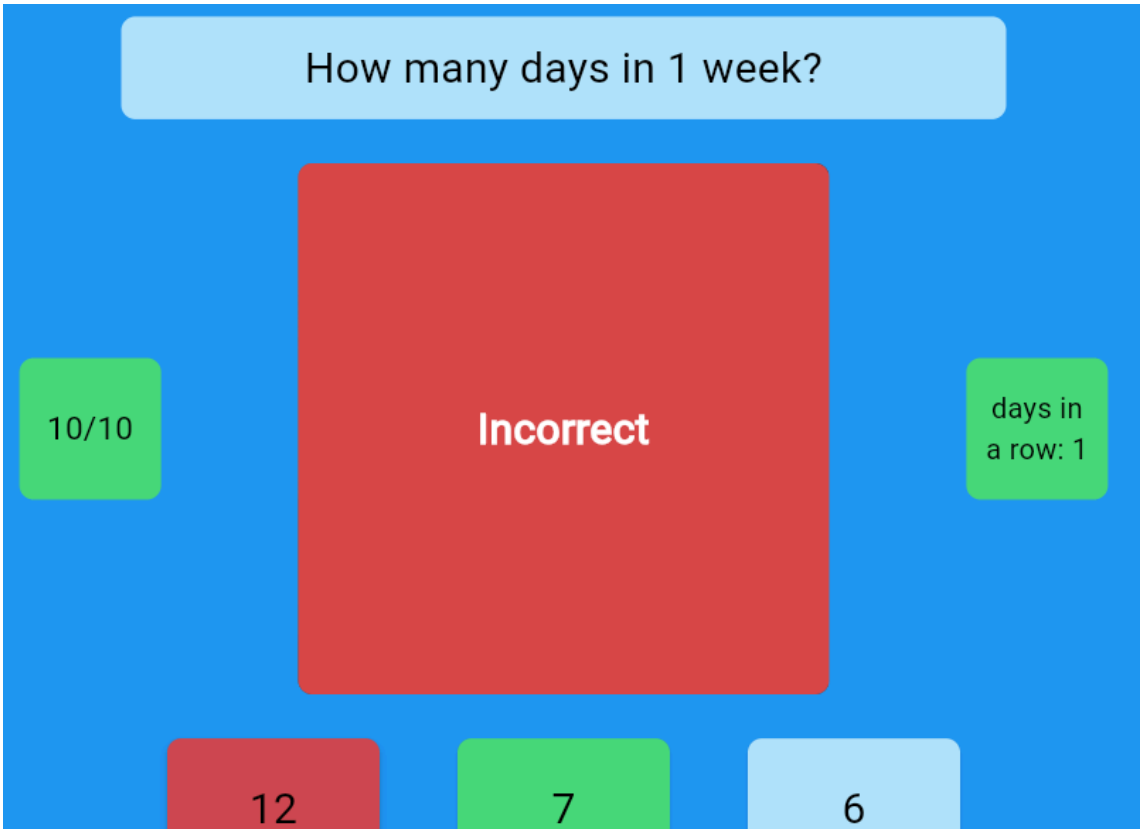


figure 5.21. Screenshot of worded question 1

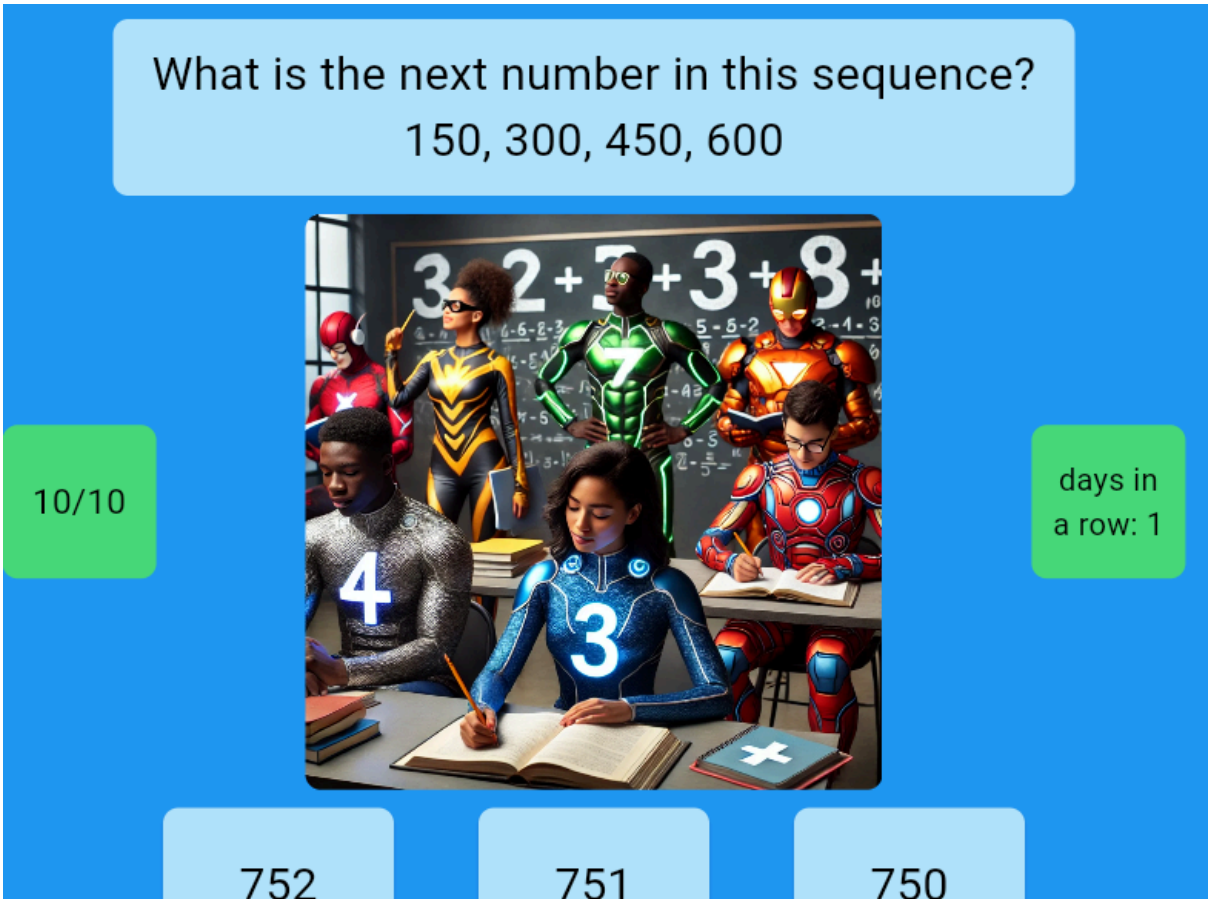


figure 5.22. Screenshot of worded question 2

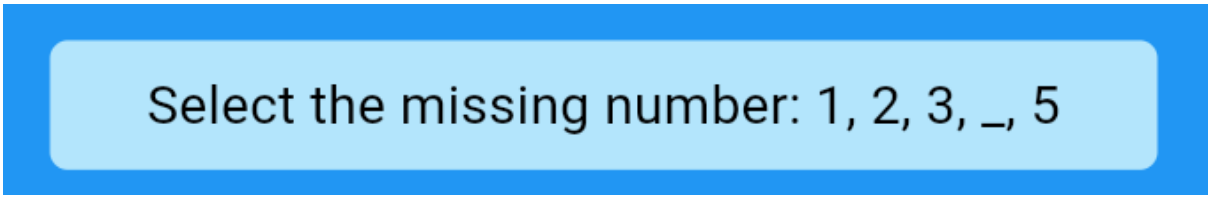


figure 5.23. Screenshot of worded question 3

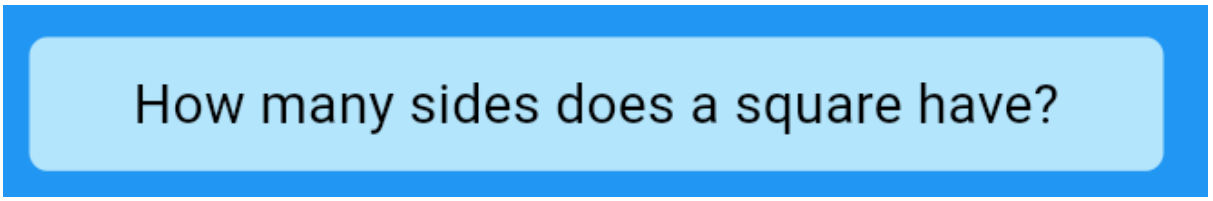


figure 5.24. Screenshot of worded question 4

Worded questions. These questions are manually written but have a chance of 0.8% to be selected to avoid repetition and as more are written this will be even less. These questions are different to the generated questions in topics and style and their purpose is to offer variety.

figure 52

Answer to the question is calculated using the generated operands and sign(in divisions case the 1st operand is calculated and the answer is already known). The correct answer is put in one of the boxes at random. To calculate the incorrect answers a random offset between -7 and +7 is added to the correct answer. Choices can't be the same as for an option to be used, it has to be different from other options. New answers are generated with every question.

figure 5.26.Screenshot of image

A random image from a file of images will be selected and displayed with each question called. This is to improve user experience.

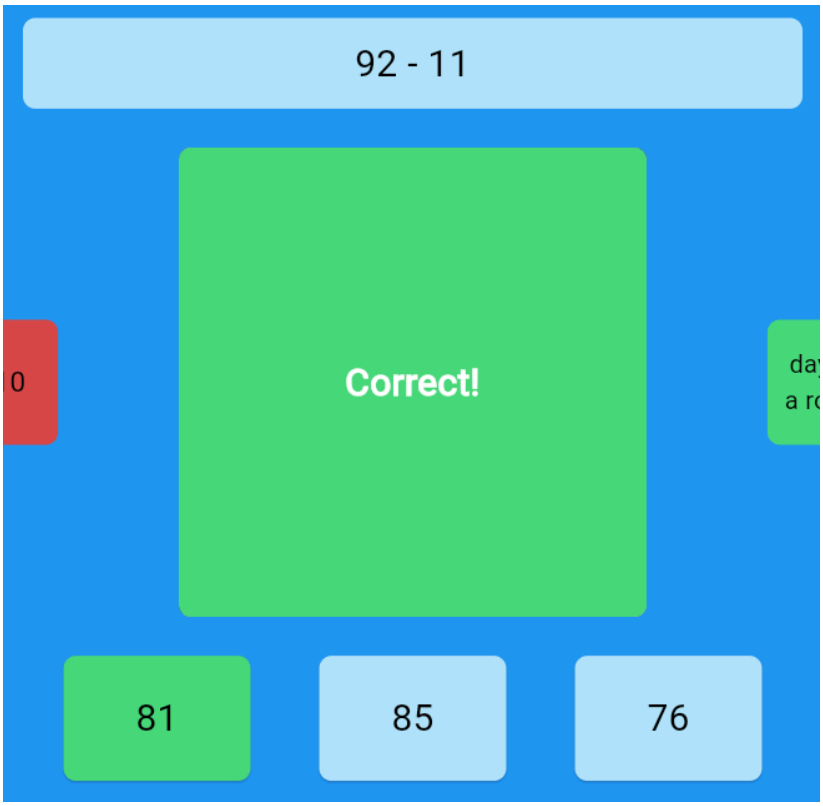


figure 5.27. Screenshot of correct answer feedback

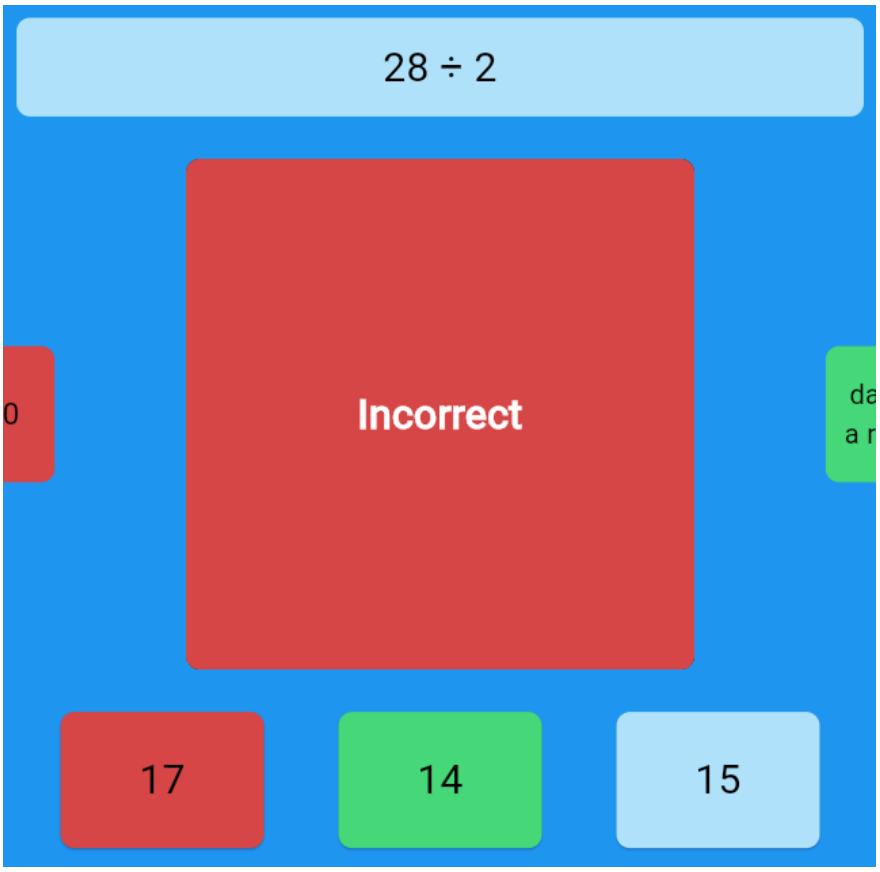


figure 5.28. Screenshot of incorrect answer feedback

Visual feedback after answering questions. This improves usability by informing users if their choice was correct and if not giving them the correct answer to help them learn from their mistake. Feedback lasts for 3 seconds or if the user presses an answer again the timer is skipped. The timer is 3 seconds as this should be long enough for all users to see and comprehend feedback. After feedback the next question arrives.

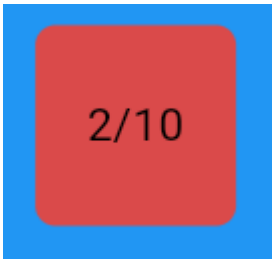


figure 5.29. Screenshot of red score/10 display

figure 5.30. Screenshot of orange score/10 display

figure 5.31. Screenshot of yellow score/10 display

figure 5.32. Screenshot of green score/10 display

Score changes colour as the user progresses with achieving 10/10. I did this to improve usability by offering more feedback. I tried to do this in a fun and colourful way that would appeal to my target audience. Score resets everyday at 12am. If the user doesn't achieve 10/10 before any given reset then days in a row will reset. The colour change is to give feedback and improve the user experience with a colourful interface. Score changes colour at 3, 6 and 10.

figure 5.33.Screenshot of days in a row display

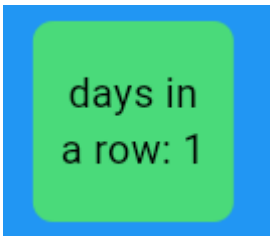


figure 5.34. Screenshot of days in a row display after increase

As soon as the score reaches 10, days in a row will update with an increment of 1. Highest days in a row will update at the same time if necessary.

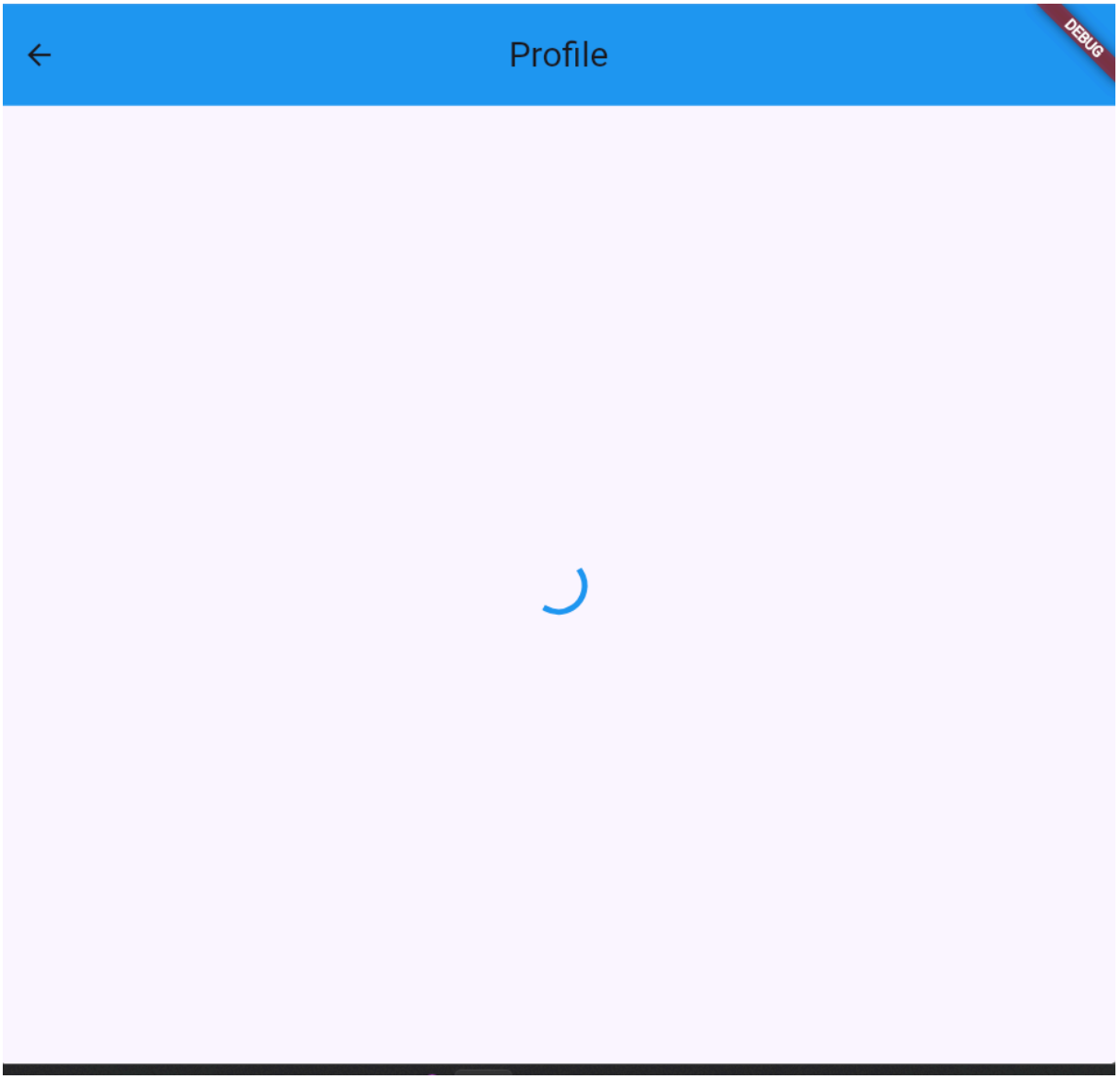


figure 5.35. Screenshot of loading indicator

Loading indicating while fetching user details on profile page. This improves usability by showing visual feedback.

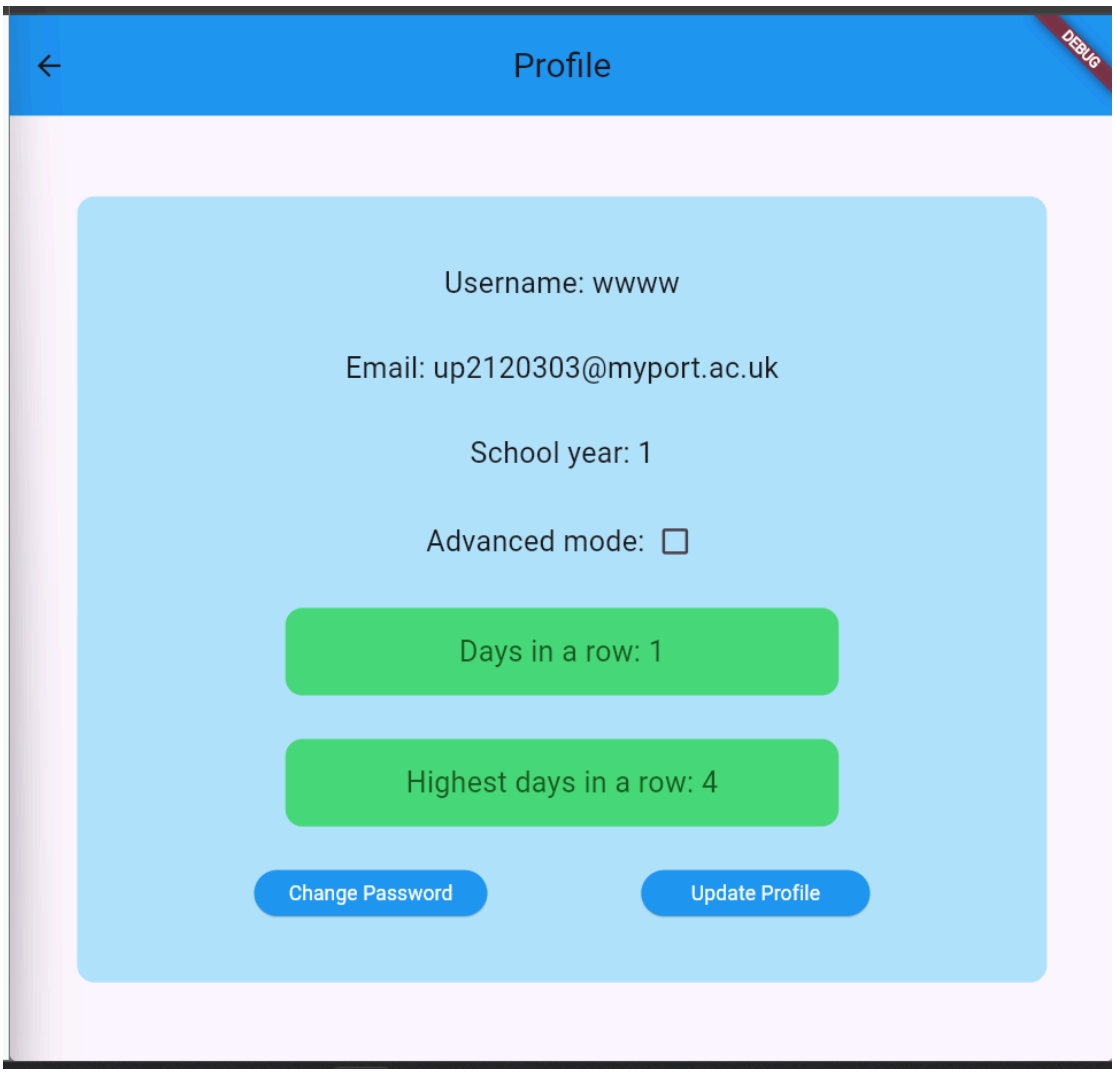


figure 5.36. Screenshot of implemented profile page

Profile page where users can view and change their details. I have made text large and only stored necessary information due to my young target audience. I have also given users the ability to customize their profile.

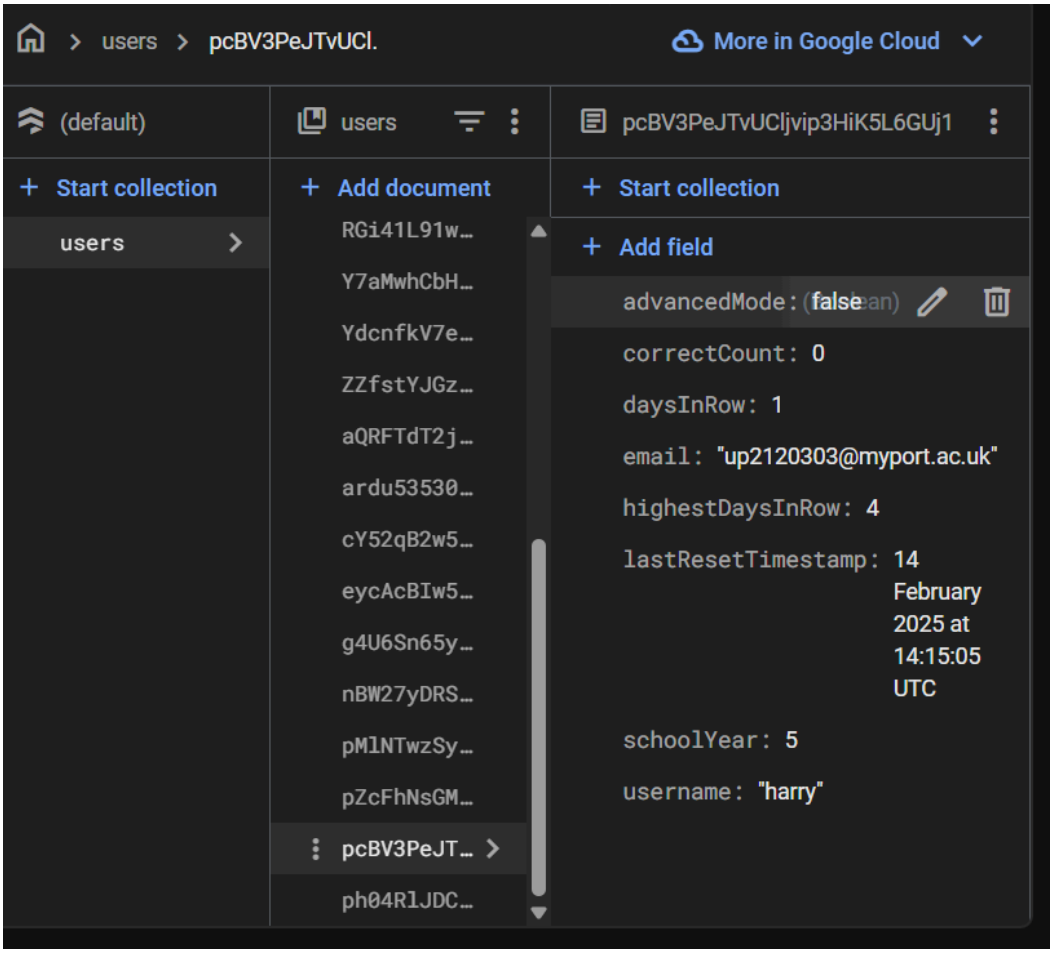


figure 5.37.Screenshot of user details from firebase

View from firebase. This is the data stored by firebase for users.

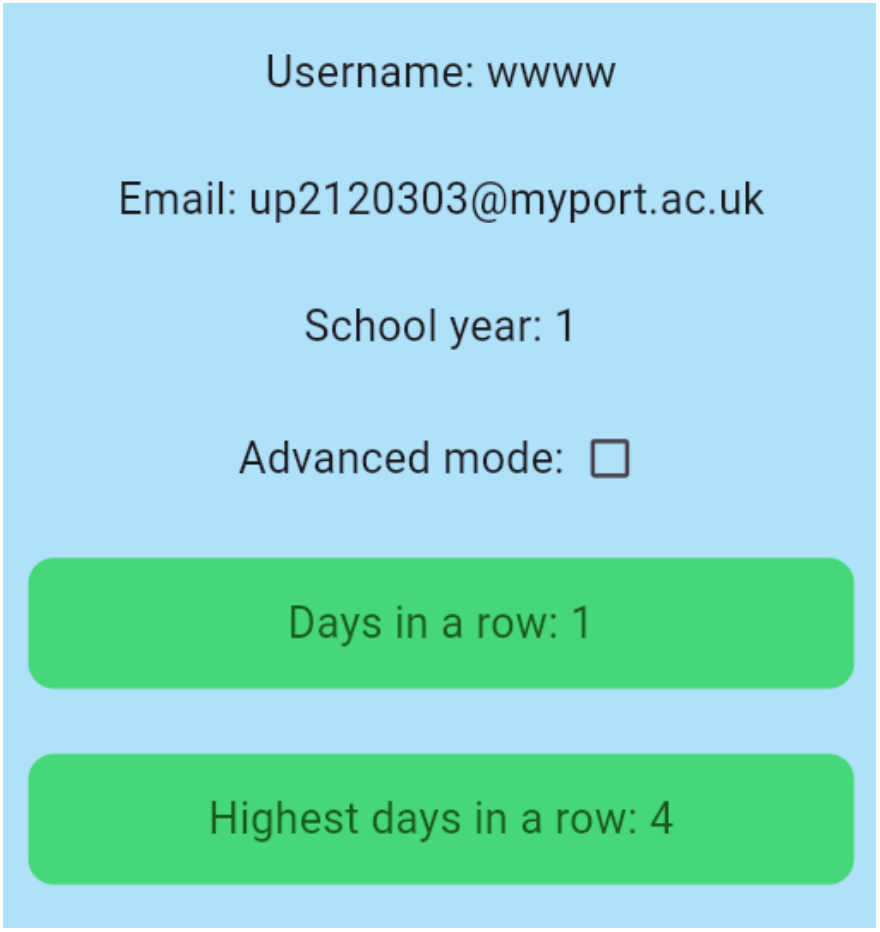


figure 5.38.Screenshot of user details

Users details. This is the only page users can see most of these values. Users can try to increase their days in a row to beat their highest days in a row. All values are called from firebase.

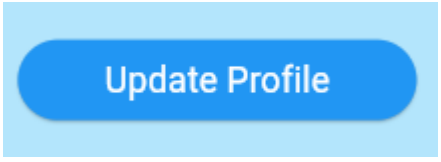


figure 5.39. Screenshot of update profile button

Button opens a form to change details.

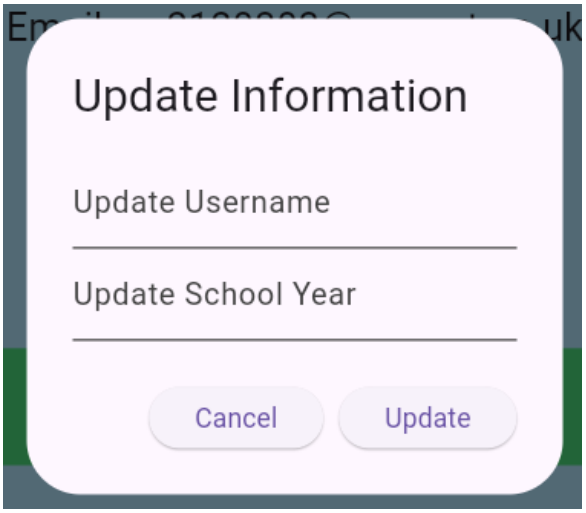


figure 5.40. Screenshot of update profile form

Form that opens for users to change details. Successfully updating form updates details on profile page and firebase.

figure 5.41. Screenshot of username error message

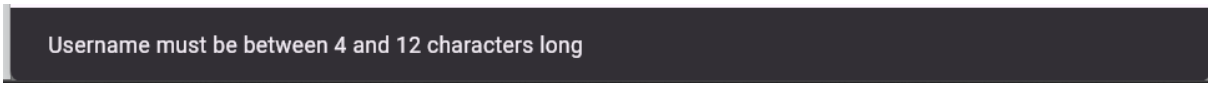


figure 5.42. Screenshot of second username error message

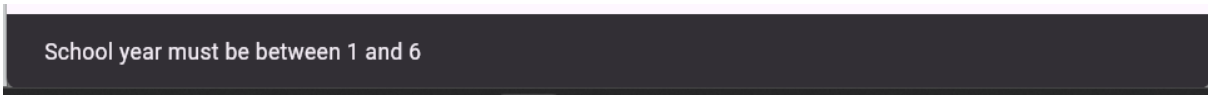


figure 5.43. Screenshot of school year error message

Error messages to help users if they are entering invalid values. This improves usability, fault tolerance and efficiency.

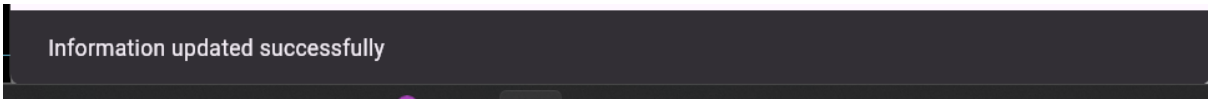


figure 5.44.Screenshot of information update message

Message confirming successful update.

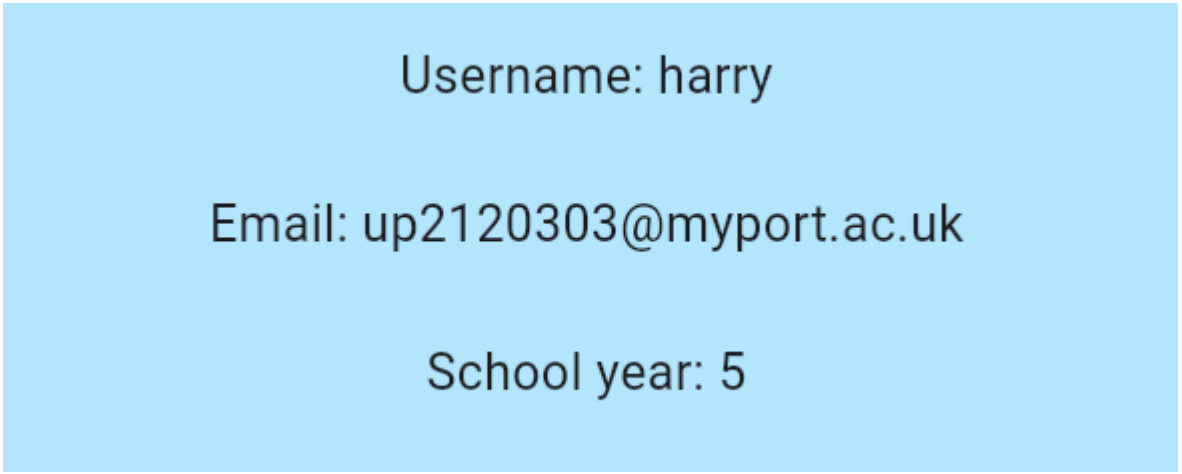


figure 5.45. Screenshot of changed details

Updated user details

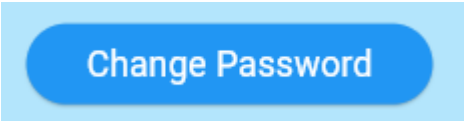


figure 5.46. Screenshot of change password button

Button opens a form to change password. Successfully updating the form updates password hash on firebase and therefore the password that lets the user login.

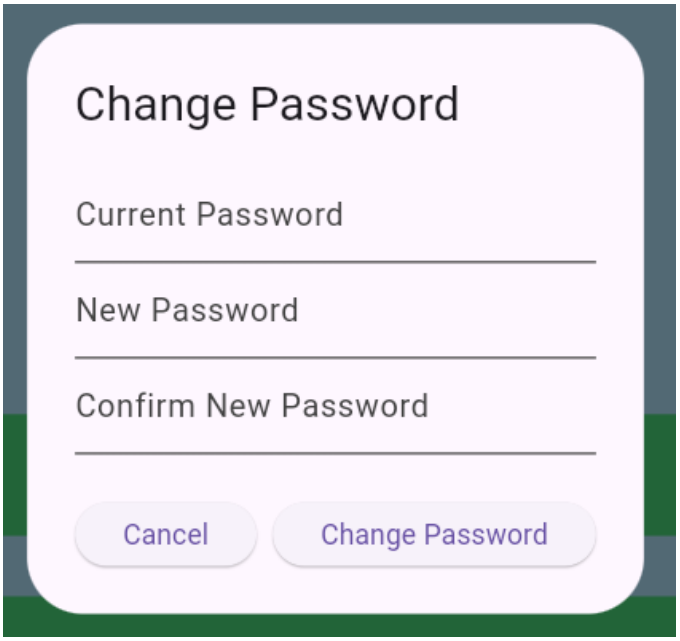


figure 5.47. Screenshot of change password form

Form that opens for users to change password.

figure 5.47. Screenshot of confirm password error message

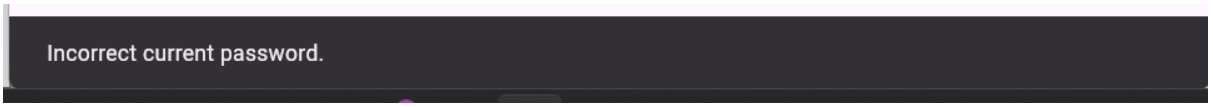


figure 5.48. Screenshot of current password error message

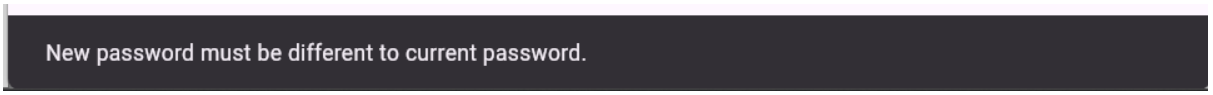


figure 5.49. Screenshot of new password error message

Error messages to help users if they are entering invalid values. This improves usability, fault tolerance and efficiency.

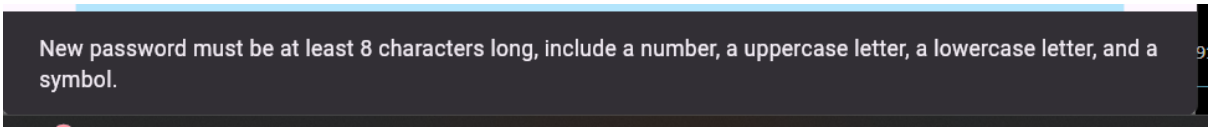


figure 5.50. Screenshot of second new password error message

This error message is displayed for a duration of 6 seconds as it is longer.

figure 5.51. Screenshot of password updated message

Message confirming successful update.

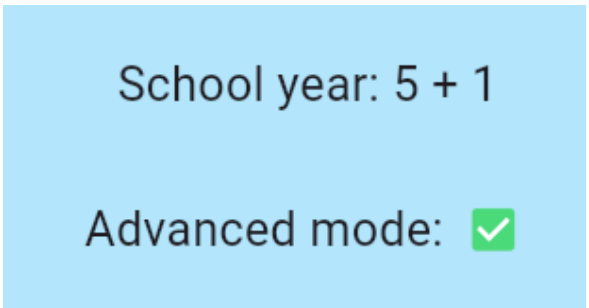


figure 5.52. Screenshot of advanced mode set to on

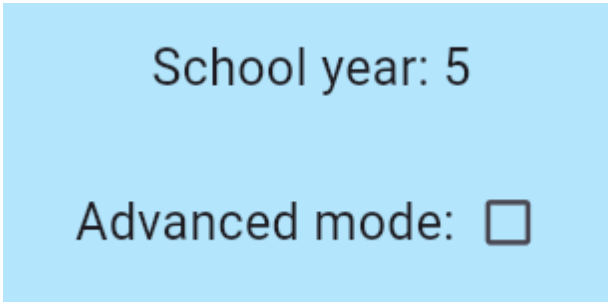


figure 5.53. Screenshot of advanced mode set to off

Advanced mode can be activated or deactivated by clicking on the square next to it. If activated, advanced mode = 1, if deactivated = 0. Advanced mode will be added to the school year to get learnage. Learnage is used when calling questions to ensure difficulty is correct. The visual prompts here with the + 1 and tick are to inform user they are using advanced mode.

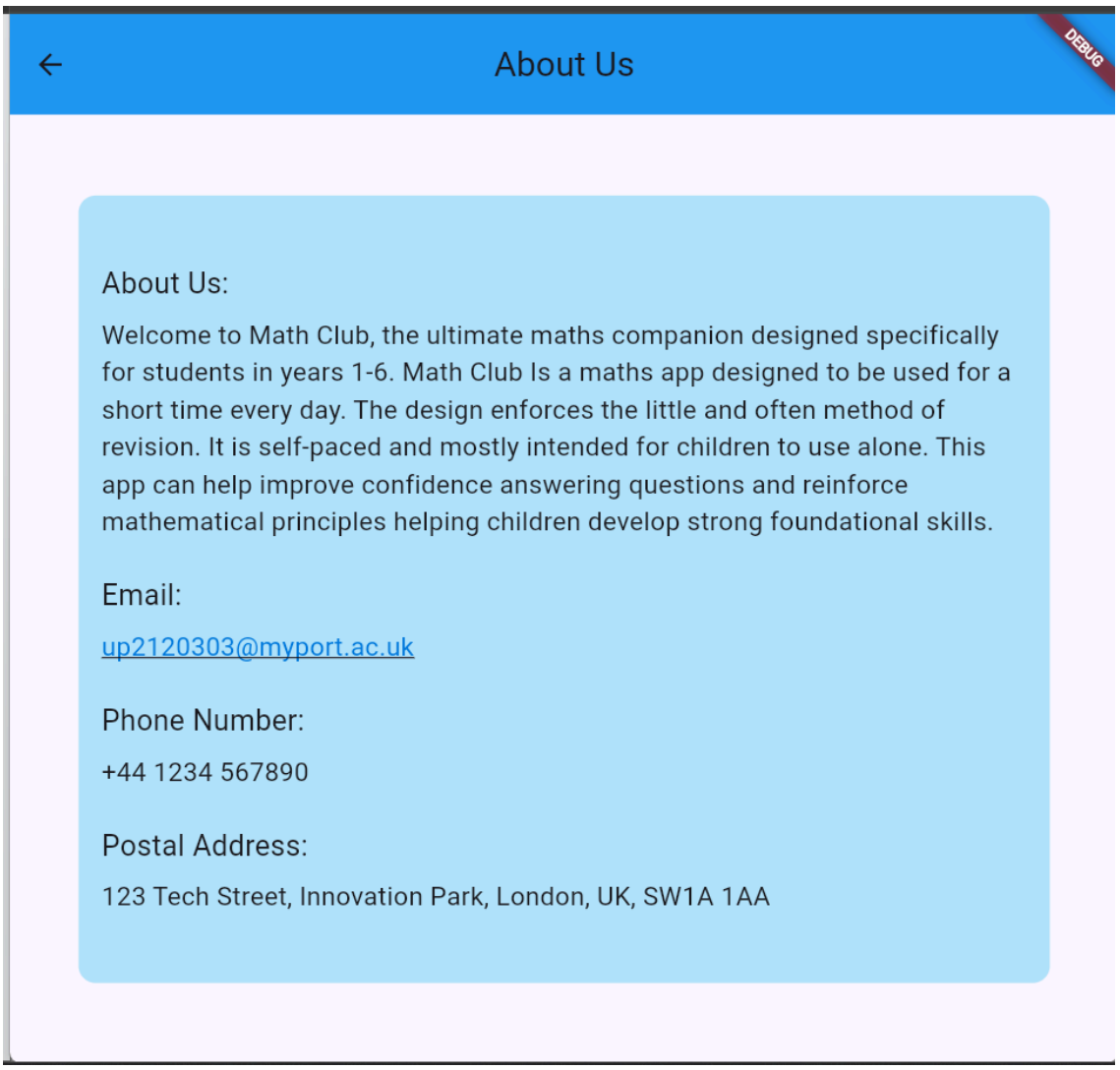


figure 5.54. Screenshot of implemented about us page

About us page that offers some information about the application and to allow users to contact administrators.

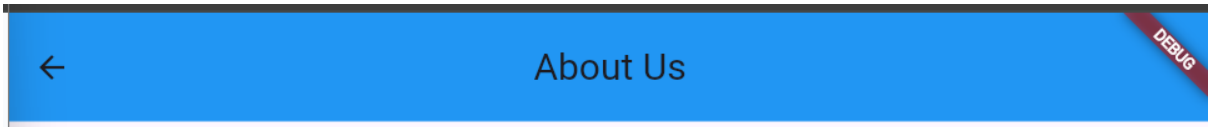


figure 5.55. Screenshot of implemented about us page banner

Uniform banner and back button for all pages excluding homepage/math club which has a dropdown menu.

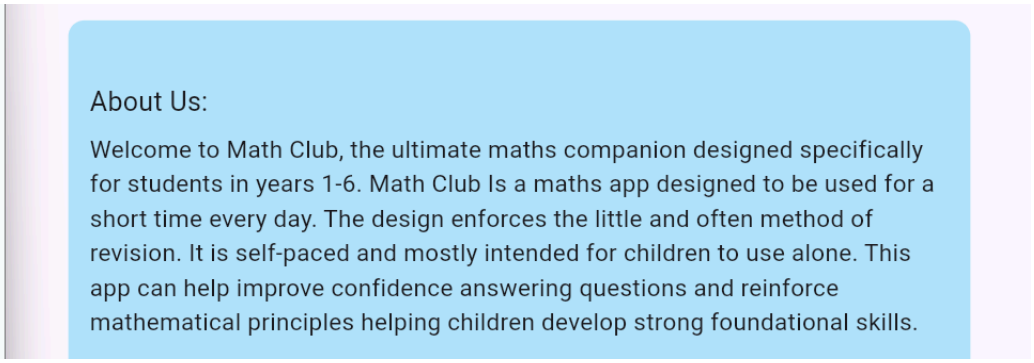


figure 5.56. Screenshot of about us paragraph

Paragraph with information about the app.

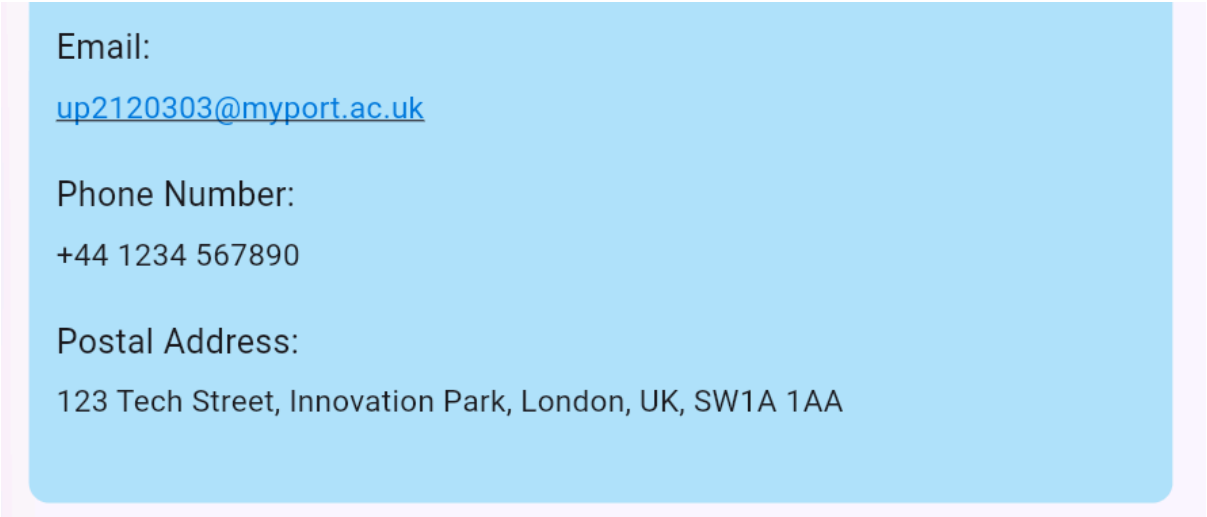


figure 5.57. Screenshot of contact details

Contact information of administrators for users to use. There is a link to email for efficiency and usability. The address and phone number are fake.

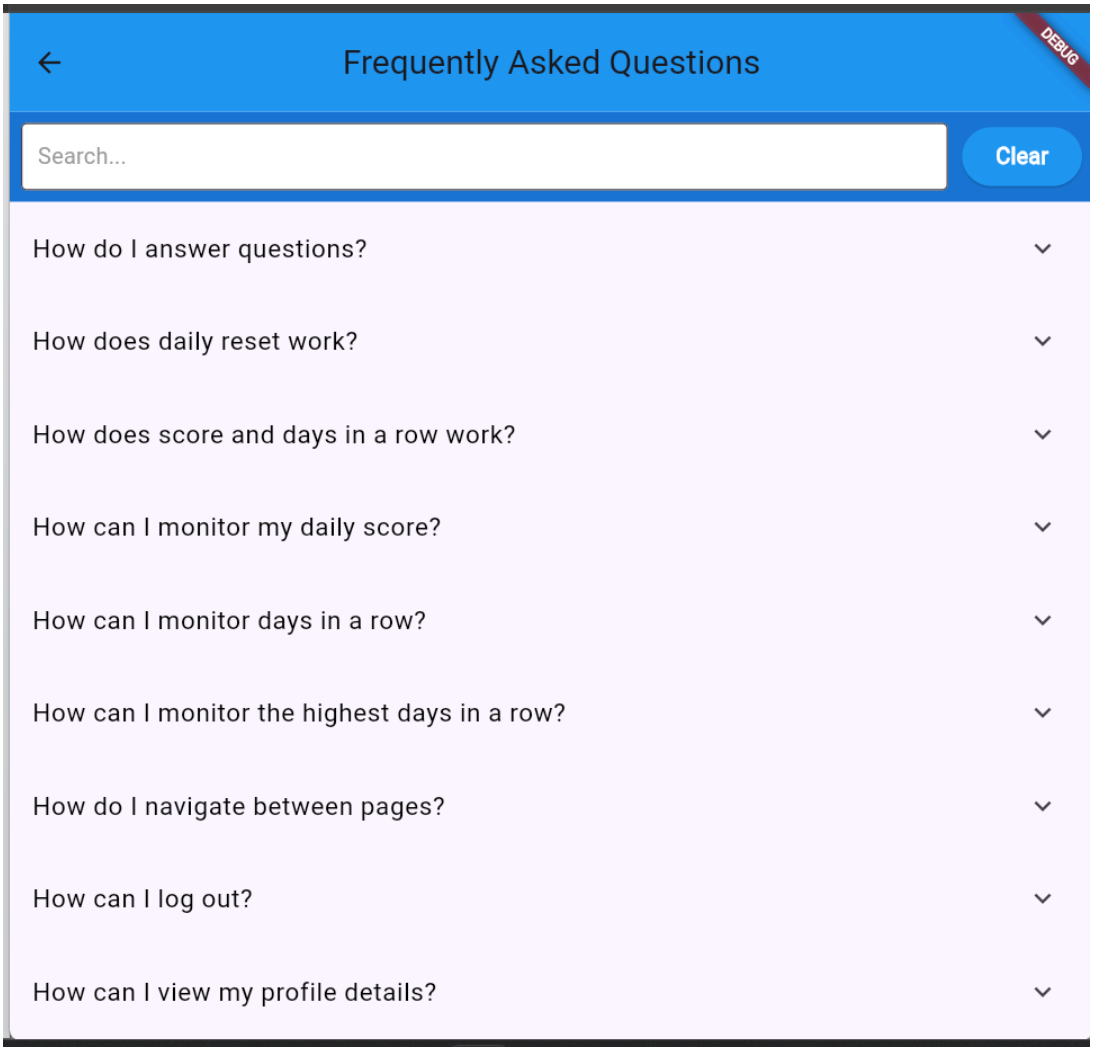


figure 5.58. Screenshot of implemented FAQ page

Searchable FAQ, each qa is clickable to see answers and the page is scrollable. This helps users who are having problems with the app and improves usability. It also helps users inform themselves about the app and its features

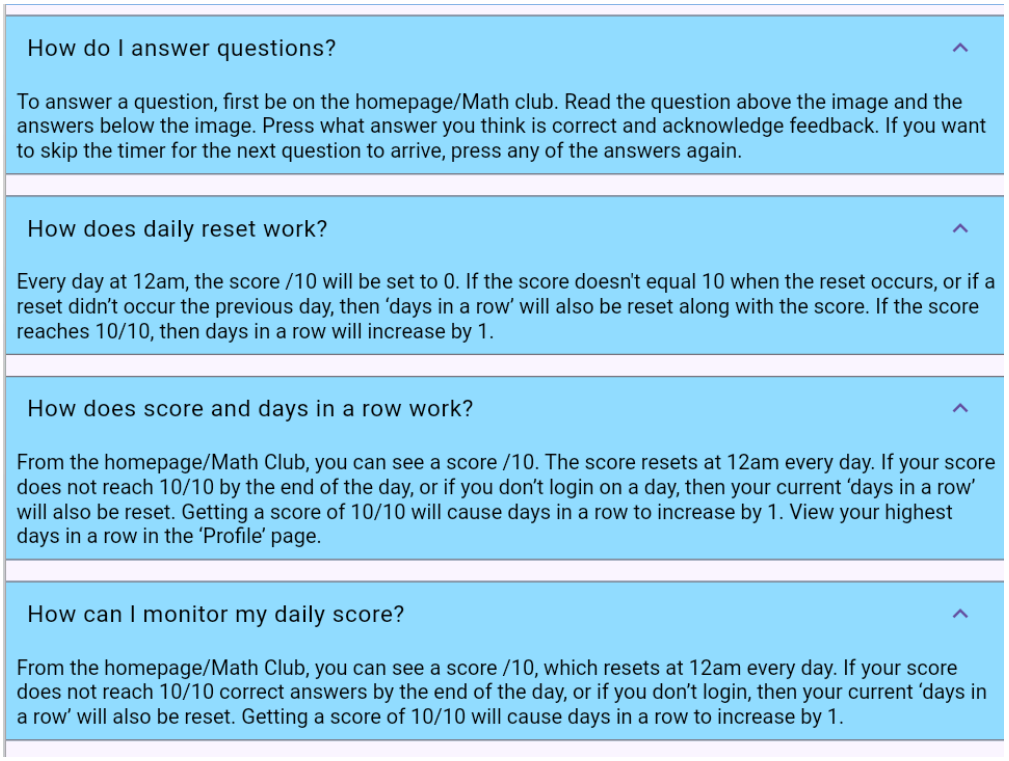


figure 5.59. Screenshot of faqs 1

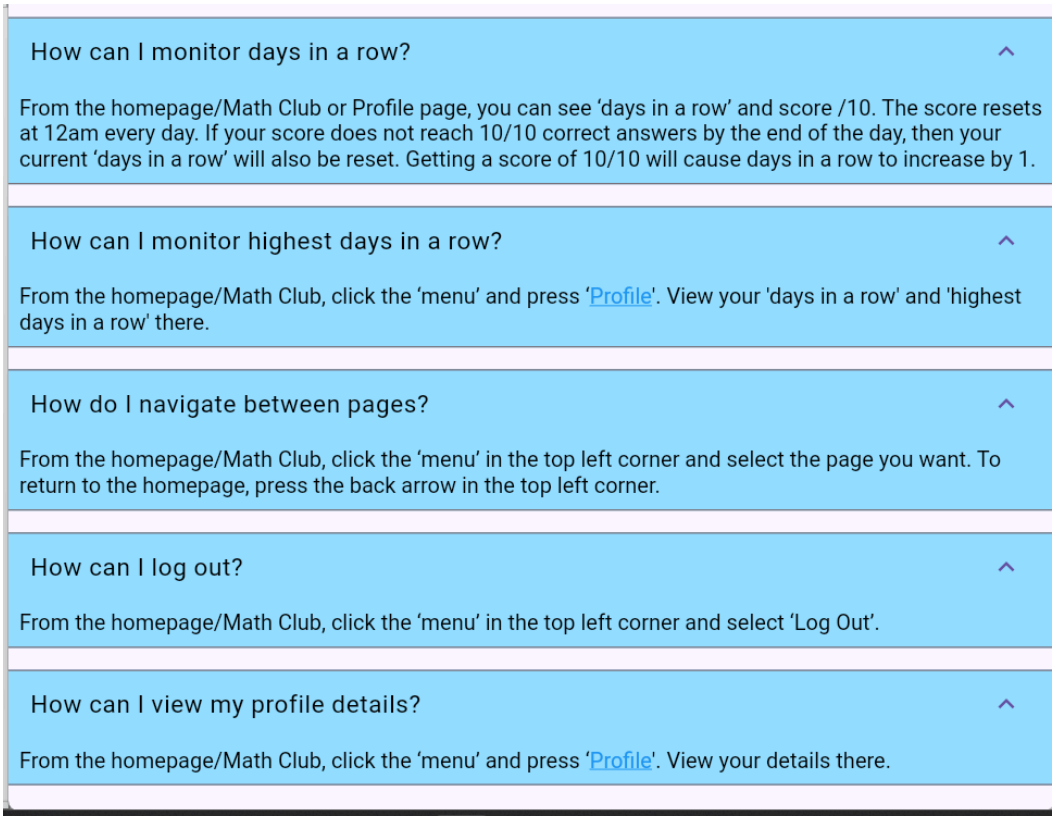


figure 5.60. Screenshot of faqs 2

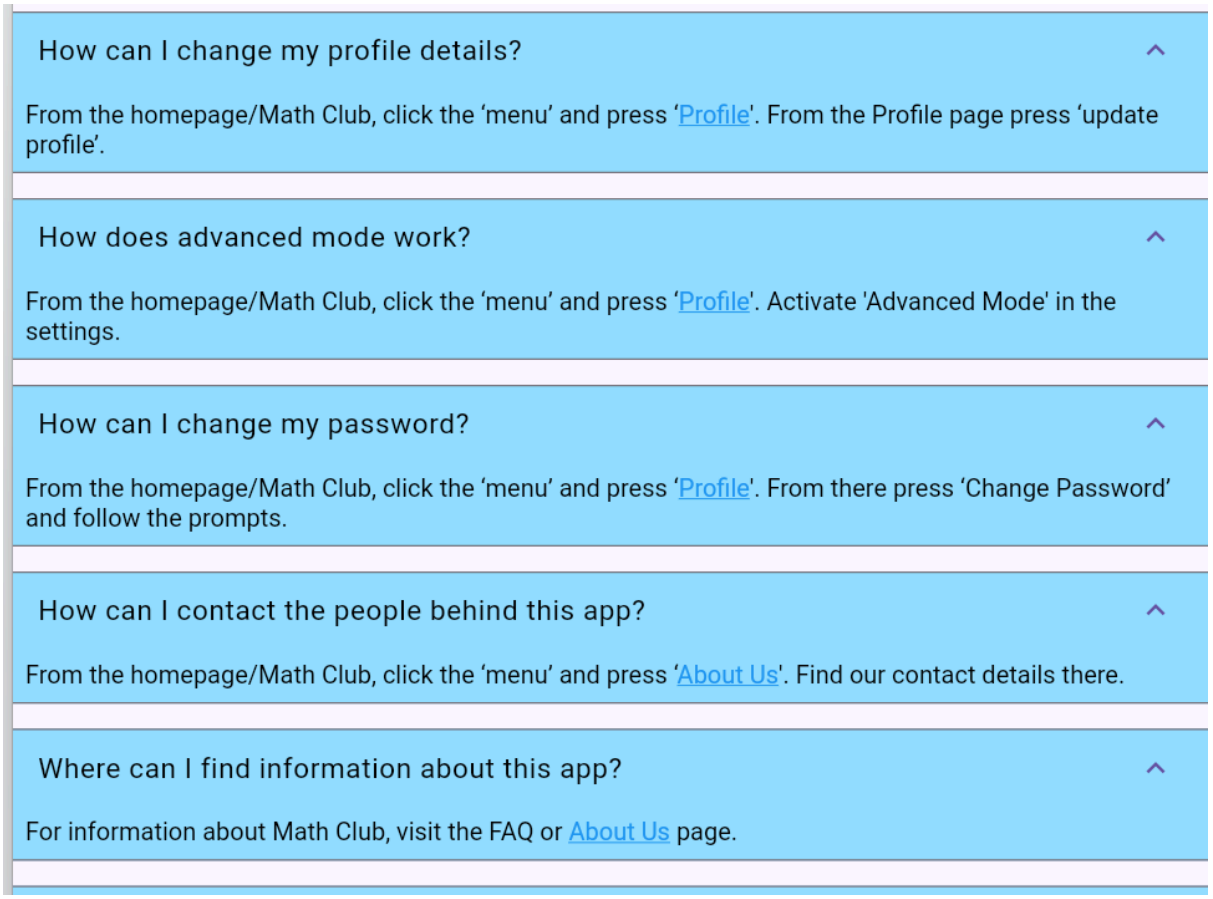


figure 5.61. Screenshot of faqs 3

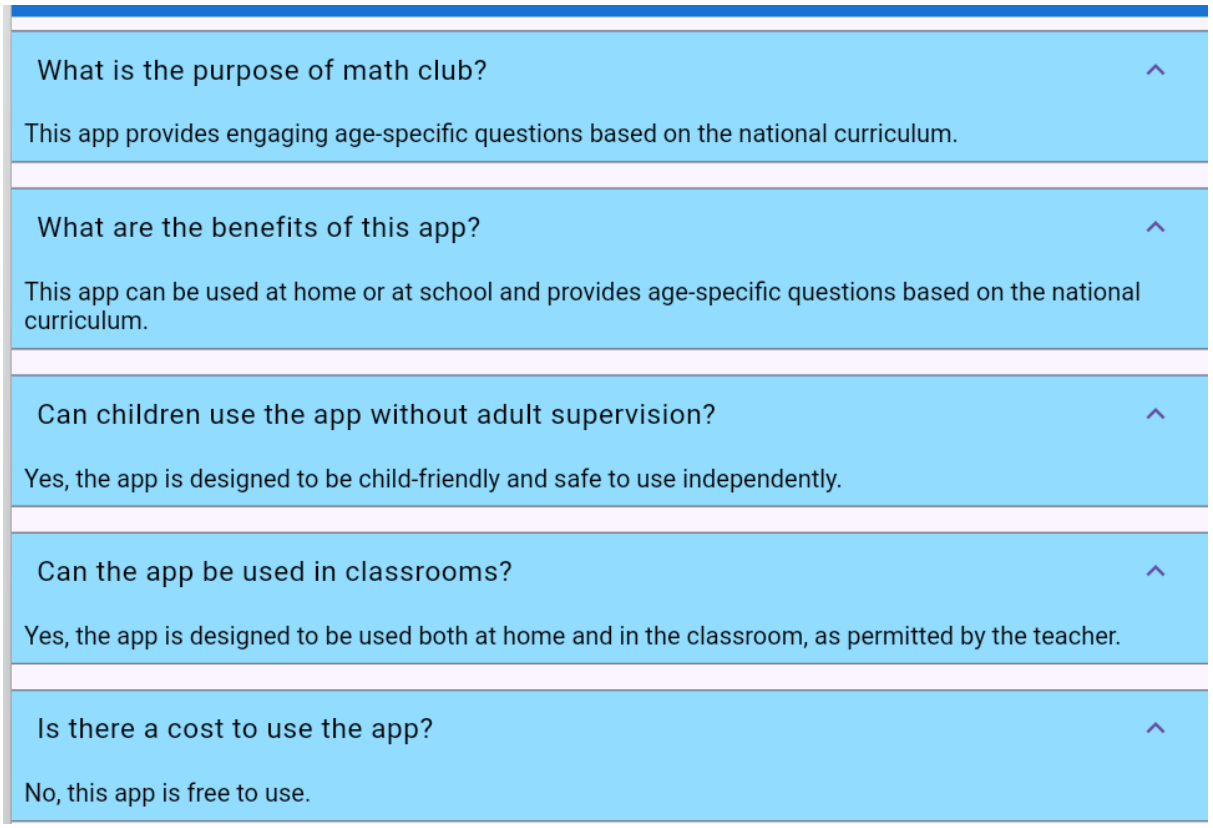


figure 5.62. Screenshot of faqs 4

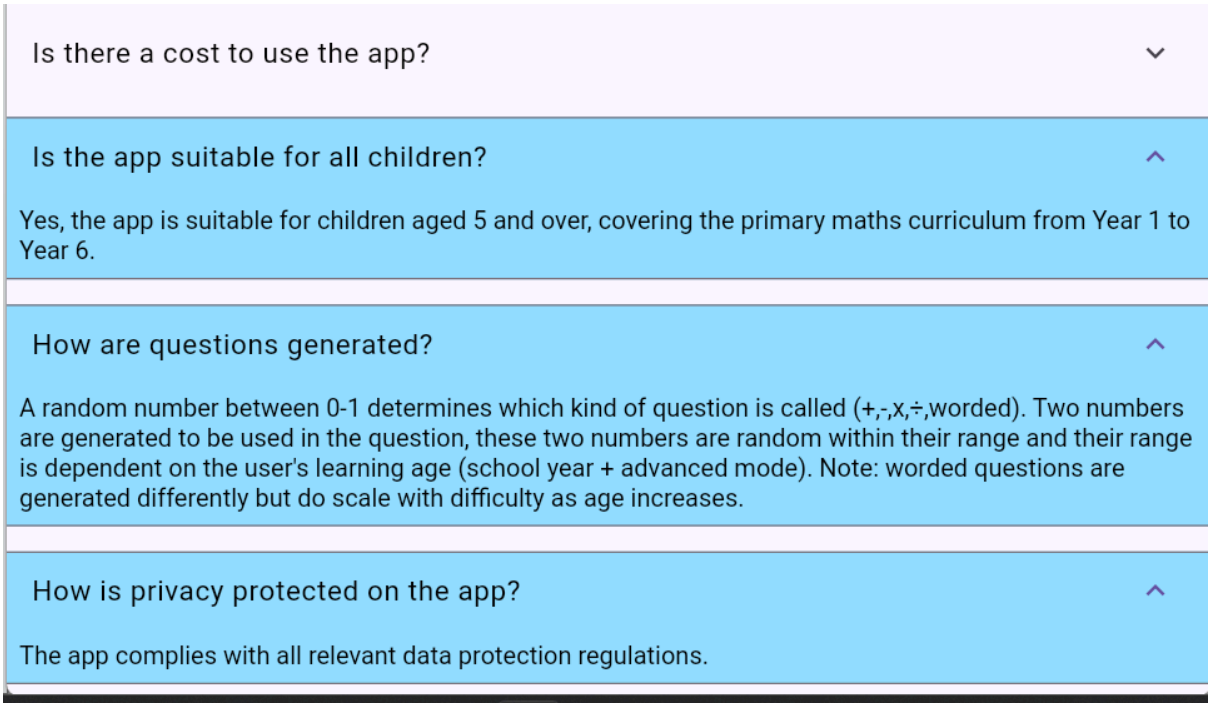


figure 5.63. Screenshot of faqs 5

All faqs after being clicked on to display answers.

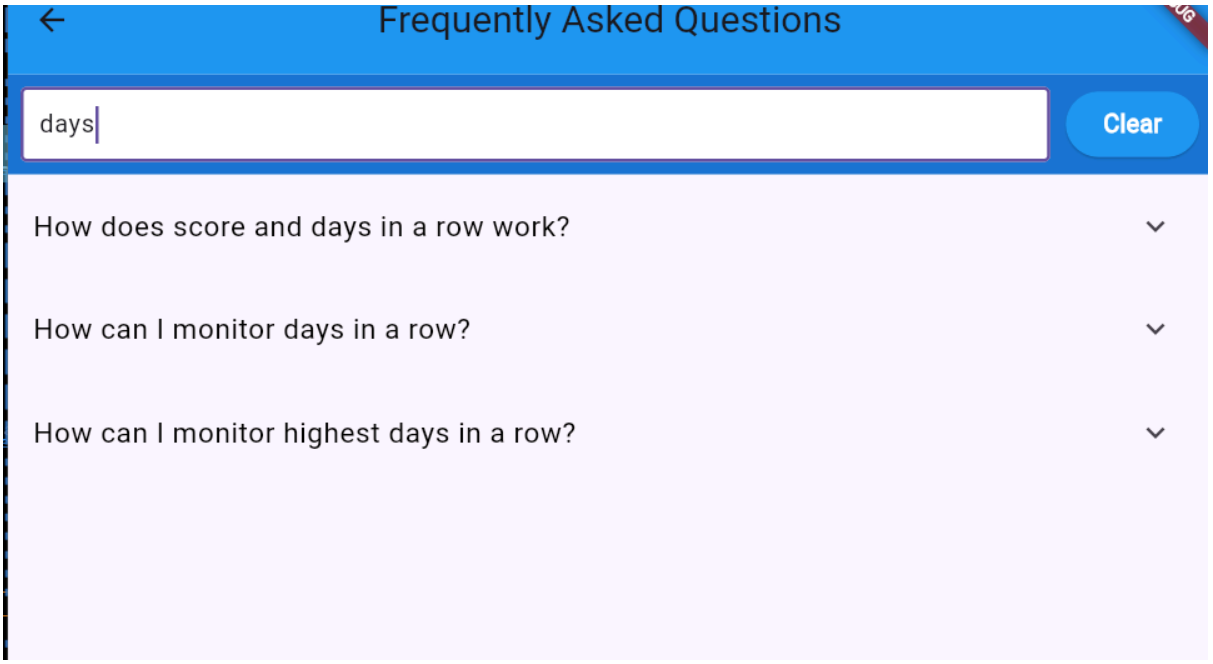


figure 5.67. Screenshot of search feature

Search features. Pressing clear would remove text ‘days’ and undo search. This helps users find qa’s and improves efficiency.

5.6. Security

Password requirements

Passwords must be at least 8 characters long, including a number, an uppercase letter, a lowercase letter, and a symbol. This forces users to choose more secure passwords. Error message explains why they can't choose a password that doesn't comply.

Firebase encryption

'Firebase services encrypt data in transit using HTTPS and logically isolate customer data.' Databases containing user data are encrypted. (Privacy and Security, 2025)

Password hashing

Firebase handles password hashing adequately. (Firebase Authentication Password, 2025)

Rules define open read/write access to database

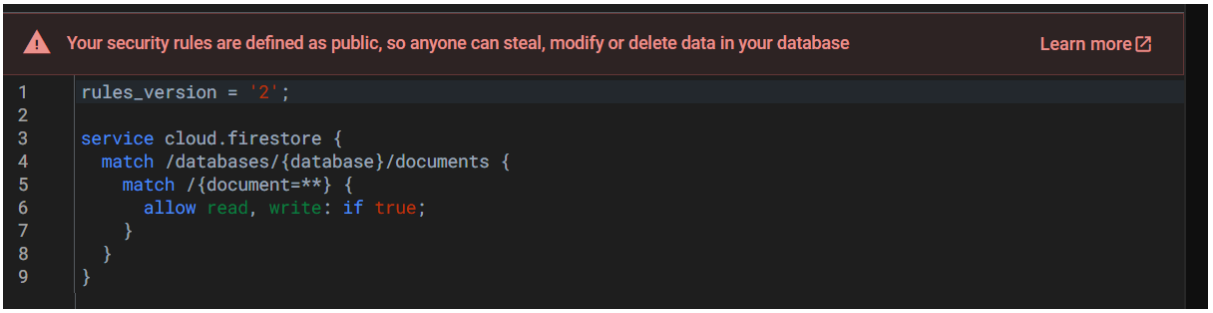


figure 5.68. Screenshot of firebase rules

This is an issue, I have experimented with different rules, by default read write was set to false. While implementing i noticed users couldn't load pages containing firebase data. To resolve this I set it to true but this is a security concern.

figure 5.69. Screenshot of updated firebase rules

These new rules check if the user has an authentic user id before granting read write access and are more secure.

Another measure I'm taking is keeping my firebase console and API key completely private.

Also when users type passwords on login and register, text is obscured.

5.7. Comparison to designs and model

Login

login is now email- This is for multiple reasons: Convenience with the firebase database, possibility to implement features that require email such as forgot password or account verification.
Error message- Unlike the design, the implemented model has an error message for login. This is to increase usability and let users know they have entered incorrect details.
Link to registration- I added this feature as it is vital for navigation

Registration

Different details- The implemented model requires password to be repeated and an email field, which are not present in the design. This is because the database requires an email field and so users don't mistype their password and get locked out of their account.
Error messages- registration has multiple error messages that correspond to different mistakes in different fields. This is a feature that wasn't in design models but is important for usability. There are a lot of fields and users need to be told what is wrong with what field if they cannot create an account.
Link to login- I added this feature as it is vital for navigation
Heading change- This page was called create account in design but register seems more appropriate.

Question/home page

better question feedback- A green/red square with correct/incorrect text covering the image offers more visually appealing answer feedback to improve the user experience
Colour changing score/10- this feature was added to improve engagement in my young target audience.
Heading change- This page was called questions in design but Maths Club seems more appropriate as it is the name of our app.

Profile page

Update profile- This feature wasn't present in design but allows users to change their username and school year. It was added to give users more control of their accounts.
Change password- This feature wasn't present in design but allows users to change their password. It was added to give users more control of their accounts
Error messages- Change details and change password have similar error messages to registration. This is to inform users why they can't make changes.
Different details- Profile page now displays an email as accounts are now linked to an email.
Heading change- This page was called account in design but profile seems more appropriate.

About us page

Phone number not linked- This is because it is a web application and the number is fake.
Layout change- simple, more appealing design to improve user experience.

Faq

Clear button- deletes text the user has typed in the search bar. This to improve error recovery if the user types the wrong thing.
Heading change- Heading now reads frequently asked questions instead of faq as some uses may not know what faq means.

General

Larger text- text size has been increased from the design to the implemented model to improve usability and readability
New colour scheme- New colour scheme is still mostly blue and uses a lot of bright colours, It is more simple, consistent, visually appealing. It also uses less white.
School year doesn't update automatically- As previously mentioned the update profile feature allows users to manually change their school year.
Intro page- basic introduction to welcome users and show app name.
More finished model- The final model includes details that aren't in designs such as meaningful text values and researched figures for questions.
Navigation- Profile, faq and about us pages have a back button instead of a dropdown menu. This is both easier for users to navigate and easier to implement.

5.8. Possible future developments

Done
Likely
Possible
Unlikely

Age update automatically- This feature could be useful but as users can update their age manually whenever they want it would be a bit of a gimmick feature. Also if users manually updated their age and then the app automatically updated their age they may not notice and answer questions on the wrong difficulty.
Forgot password on login- This would help users if they can't get into their account.
Verify account- This could be used to implement parental consent.
Login with username or email- This feature could give users more options when logging in and username may be convenient as it is generally shorter than an email address.
Multiple error messages for login- Different error messages indicating if the email isn't registered or the password is wrong could help users if they can't login.
More worded questions
More images and questions- This would make the overall experience less repetitive. Different ideas could be explored for types of questions.
Squared questions
Cubed questions
3 number addition
Square root questions
Tidy menu so about us fits on 1 line- More appealing for users.

5.9. Late additions

More question types added and probabilities changed
Before changes each question type had a 20% chance to be selected. These additional question types will change probabilities around as there probabilities will deduct from related topics. For example ‘route of’ question type has a 4% chance which is deducted from divisions 20%.

New probabilities of each question type on each difficulty will be (new question types in red)

Year 1:
40% addition
40% subtraction
20% worded questions

Year 2:
15% addition
5% 3 number addition
20% subtraction
20% multiplication
20% division
20% worded questions

Year 3:
15% addition
5% 3 number addition
20% subtraction
20% multiplication
20% division
20% worded questions

Year 4:
15% addition
5% 3 number addition
20% subtraction
20% multiplication
20% division
20% worded questions

Year 5:
15% addition
5% 3 number addition
20% subtraction
12% multiplication
4% squared
4% cubed
15% division
5% route of
20% worded questions

Year 6:
15% addition
5% 3 number addition
20% subtraction
12% multiplication
4% squared
4% cubed
15% division
5% route of
20% worded questions

Advanced year 6:
15% addition
5% 3 number addition
20% subtraction
12% multiplication
4% squared
4% cubed
15% division
5% route of
20% worded questions

```
else if (learnage == 5) {
    double randomNumber1 = random.nextDouble();
  int result1, result3, result4;
  String result2;

  if (randomNumber1 < 0.2) {
    result2 = "-";
    result1 = random.nextInt(5000) + 1;
    int upperLimit = min(result1, 5000);
    result3 = random.nextInt(upperLimit) + 1;
    correctAnswer = result1 - result3;
    question = "$result1 $result2 $result3";
  } else if (randomNumber1 < 0.32) {
    result2 = "x";
    result1 = random.nextInt(12) + 1;
    result3 = random.nextInt(100) + 1;
    correctAnswer = result1 * result3;
    question = "$result1 $result2 $result3";
  } else if (randomNumber1 < 0.36) {
    result2 = "2";
    result1 = random.nextInt(14) + 1;
    correctAnswer = result1 * result1;
    question = "$result1 $result2 ";
  } else if (randomNumber1 < 0.4) {
    result2 = "3";
    result1 = random.nextInt(5) + 1;
    correctAnswer = result1 * result1 * result1;
    question = "$result1$result2 ";
  } else if (randomNumber1 < 0.55) {
    result2 = "+";
    result1 = random.nextInt(5000) + 1;
    result3 = random.nextInt(5000) + 1;
    correctAnswer = result1 + result3;
    question = "$result1 $result2 $result3";
```

figure 5.70.Screenshot of question generation algorithm for year 5

```
  } else if (randomNumber1 < 0.60) {
    result2 = "+";
    result1 = random.nextInt(1000) + 1;
    result3 = random.nextInt(1000) + 1;
    result4 = random.nextInt(1000) + 1;
    correctAnswer = result1 + result3 + result4;
    question = "$result1 $result2 $result3 $result2 $result4";
  } else if (randomNumber1 < 0.75) {
  result1 = random.nextInt(12) + 1;
  correctAnswer = random.nextInt(100) + 1;
  result3 = result1 * correctAnswer;
  result2 = "÷";
  question = "$result3 $result2 $result1";
  } else if (randomNumber1 < 0.80) {
  correctAnswer = random.nextInt(10) + 1;
  result1 = correctAnswer * correctAnswer;
  result2  = "route of";
  question = "$result2 $result1";
 } else {
int wordedQuestion = random.nextInt(10);
```

figure 5.71.Second screenshot of question generation algorithm for year 5

Updated algorithm to generate questions and answers with new probabilities (learnage/year 5)

3 number addition

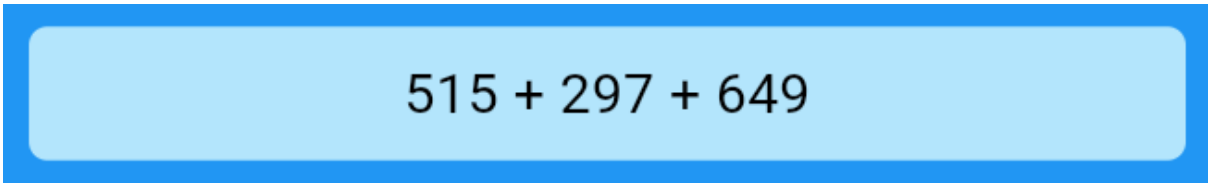


figure 5.72. Screenshot of 3 number addition question

This question type asks users to add together 3 generated numbers. The upper range of these numbers change with the users school year and users will get this question type from year 2. This question type and its ranges were based on my research (DfE, 2013).

Upper limits of three numbers to be used:
Year 2: 10, 10, 10
Year 3: 100, 50, 50
Year 4: 100, 100,100
Year 5: 1000, 1000, 1000
Year 6: 1000, 1000, 1000
Advanced year 6: 1000, 1000, 1000

Squared

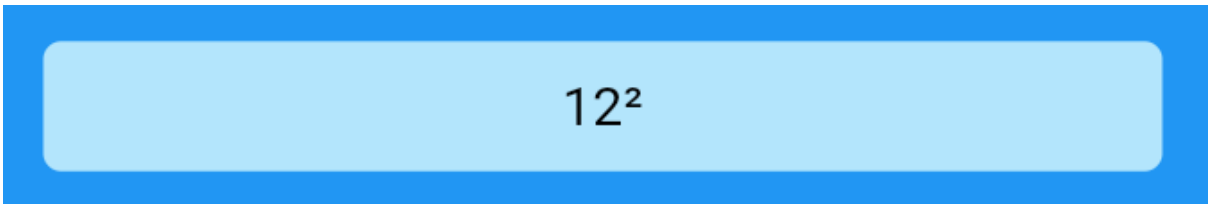


figure 5.73. Screenshot of squared question

This question type presents the user with a number squared and they have to calculate the answer. The upper range of this number changes with the user's school year and users will get this question type from year 5. This question type and its ranges were based on my research (DfE, 2013).

Upper limits of number to be squared
Year 5: 14
Year 6: 15
Advanced year 6: 20

Cubed

figure 5.74. Screenshot of cubed question

This question type presents the user with a number cubed and they have to calculate the answer. The upper range of this number changes with the user's school year and users will get this question type from year 5. This question type and its ranges were based on my research (DfE, 2013).

Upper limits of number to be squared
Year 5: 5
Year 6: 10
Advanced year 6: 12

Root of

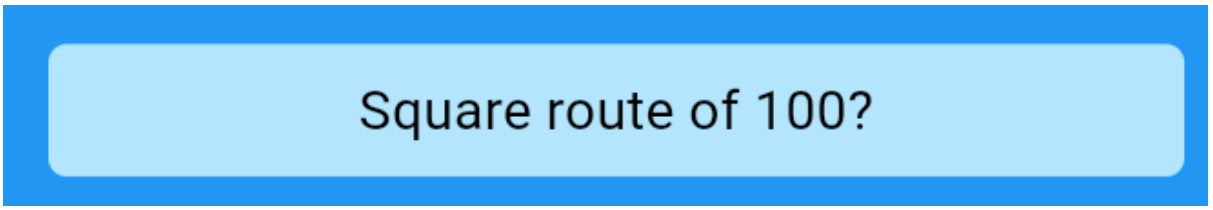


figure 5.75. Screenshot of square root question

This question type presents the user with a number and they have to calculate the square root. The upper range of this root changes with the user's school year and users will get this question type from year 5. This question type and its ranges were based on my research (DfE, 2013).

Upper limits of answer
Year 5: 10
Year 6: 12
Advanced year 6: 15

Menu fix

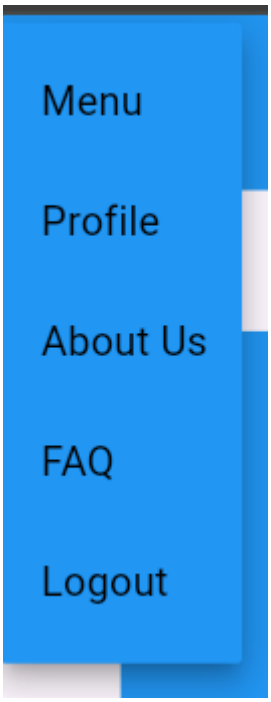


figure 5.76. Screenshot of fixed menu

About us text on 1 line to make ui more visually appealing

Chapter 6: Testing

6.1. Introduction

Multiple testing procedures have been performed including functionality testing, performance testing, compatibility testing, correctness check, grammatical correctness check and a usability study. There are 92 programmed tests as well as multiple non programmed tests.

6.2. Coded functional test plan with results

Some tests mock firebase features for test users, however the features being tested would use firebase in real cases. Therefore to ensure consistency and functionality every functional test has been ran manually before being implemented.

Introduction

Test name	Tests	Inputs	Results	Notes
container	Container is present and contains correct texts: ‘Welcome to Math Club’ and ‘Click anywhere to continue’. Clicking on container directs users to login page	Container pressed	Test passed	-
Back	Back on intro shouldn’t navigate users to a new page they should remain on intro page	Operating system back button pressed	Test passed	Back button not present so built in back function used

figure 6.1. Coded functional test plan and results: introduction page

Login

Test name	Tests	Inputs	Results	Notes
Appbar	Banner is present with correct properties and contains text: ‘login’	null	Test passed	-
Typable fields	Typable fields for email and password are present and working and	Email: test@example.com Password: password123	Test passed	-

	password text is obscured on the user interface.			
Incorrect login, Successful login,	Button containing text ‘Login’ is present and on press takes and validates users input, It will either navigate user to question page or display an error message	‘Login’ button pressed after various inputs	Test passed	fakeUser.dart used to mock validation services using signInWithEmail
Successful login	Valid input directs user to question page (homepage/math club)	Email: 27946228@example.com Password: Test123! ‘Login’ button pressed	Test passed	
Incorrect login	Invalid input displays error message: ‘Incorrect email or password’ and user remains on login page	Login button pressed Email: gicepa7720@dwriters.com Password: wrongpassword Login button pressed Email: invalid Password: 11111111!Aa Login button pressed	Test passed	2 instances First is no input, second is valid email with the wrong password
Register	Button containing text: ‘Don’t have an account? Register’ is present and on press redirects users to registration	‘Don’t have an account? Register’ button pressed	Test passed	-
Back	Back on login should navigate users to intro	Built in web back button performed	Test passed	Back button removed from appbar so built in back function used

figure 6.2. Coded functional test plan and results: login page

About us

Test name	Test	Inputs	Results	Notes
Appbar	Banner is present with correct properties and contains text: ‘login’	null	Test passed	-
Container	Container is present with correct properties	null	Test passed	-
Text	Paragraph with correct text is present: ‘welcome to Math Club, the ultimate maths companion designed specifically for students in years 1-6. Math Club Is a maths app designed to be used for a short time every day. The design enforces the little and often method of revision. It is self-paced and mostly intended for children to use alone. This app can help improve confidence answering questions and reinforce mathematical principles, helping children develop strong foundational skills.’	null	Test passed	-
Phone Number	Correct phone number: ‘+44 1234 567890’ is present	null	Test passed	-
Postal Address	Correct address: ‘123 Tech Street, Innovation Park, London, UK, SW1A 1AA’ is present	null	Test passed	-

Email	Clicking email link opens page to mail ‘up2120303@myport.ac.uk’	Link/email text pressed	Test passed	‘mockEmail.dart’ is used to mock this feature
Back	Back button is functional and navigates users to question page	Back icon pressed	Test passed	removed from appbar so built in back function used

figure 6.3. Coded functional test plan and results: About us page

Faq

Test name	Test	Inputs	Results	Notes
Appbar	Banner is present with correct properties and contains text: ‘Frequently Asked Questions’	null	Test passed	-
Search	There are 22 questions. After typing q in the search bar there should be 2 questions. After pressing clear their should be 22 questions	Scroll down Scroll up Search: ‘q’ ‘Clear’ button pressed Scroll down	Test passed	-
faqs	Click every faq and check the text for questions and answers is correct.	Every question pressed Scroll down	Test passed	-
Links	Links on answers should go to the page with name corresponding to the phrase with the link After navigating from faq to profile or about us pages and pressing back icon users go back to faq page	Navigate to pages Press back button	Test passed	Testing framework struggles to find the linked text as it is embedded in other text so I manually navigated to the destinations.
Back	Back button is functional and navigates users to question page	Back icon pressed	Test passed	-

figure 6.4. Coded functional test plan and results: faq page

Register
New rules:

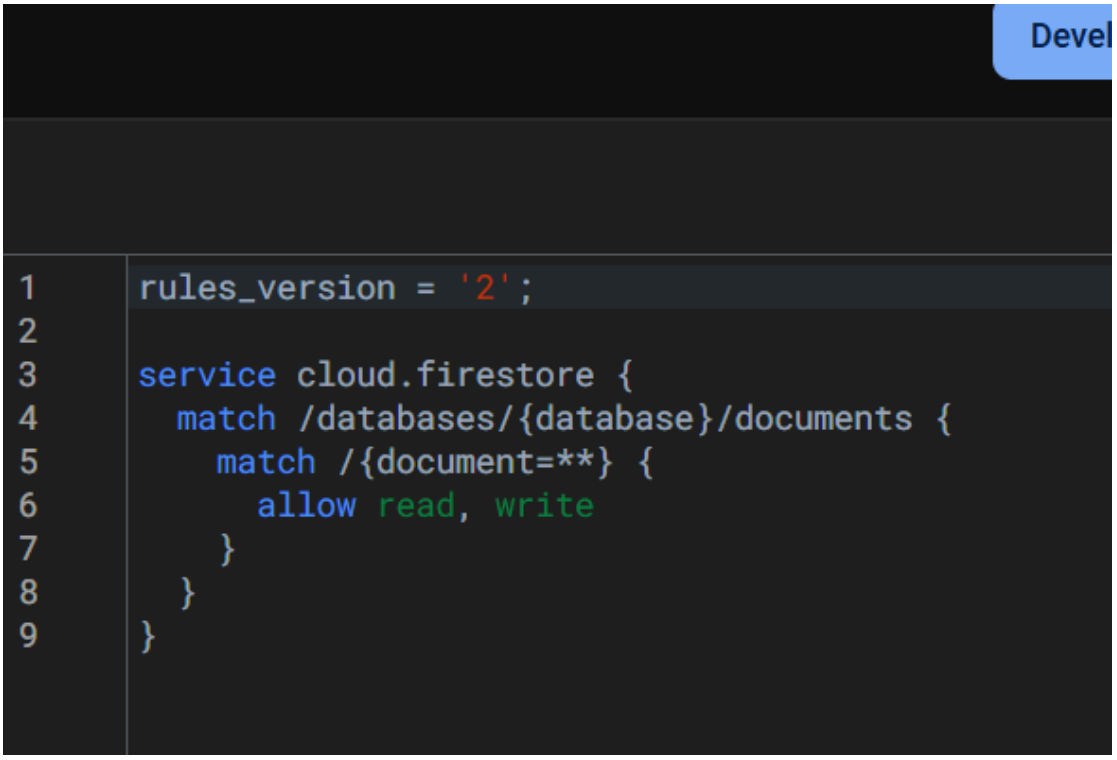


figure 6.5. Screenshot of updated firebase rules

It came to my attention that the firebase rules I had only let users with a user ID write the database. Before registration users don't have a user ID yet, This meant accounts couldn't be created. I changed my rules to fix this. These rules don't impose restrictions on users but may need to be evaluated.

Test name	Tests	Inputs	Results	Notes
Appbar	Banner is present with correct properties and contains text: 'register'	null	Test passed	-
login	Button containing text: 'Already have an account? Login' is present and on press redirects users to login	'Already have an account? Login' button pressed	Test passed	-
Typable fields	Typable fields for email, username, school year, password and confirm password are present and working and both password texts	Email: test@example.com Username: testuser School year: 3 Password: password123	Test passed	-

	are obscured on the user interface.	Confirm password: password123		
Register, unique email, unique username, username length, password, confirm password, school year, email,	Button containing text ‘Register’ is present and on press takes and validates users input, It will either navigate user to question page or display an error message	‘Register’ button pressed after various inputs	Test passed	‘fakeUser’ used to mock validation services, create accounts, update mock databases, check if inputs are in mock databases,
Register	Valid input directs the user to the homepage/math club but many conditions have to be met. Upon successful registration email is saved to test_registered_emails.json and username is saved to test_registered_usernames.json	Email: (unique and random email not in test_registered_emails.json) Username: (unique and random username not in test_registered_usernames.json) School year: 3 Password: test123! Confirm password: test123! ‘Register’ button pressed	Test passed	Mock emails and usernames are made by generating a random 8 digit number and combining it with a string: userx for username and x@example.com with x being the random number. Before these fields are used in registration a check is performed that username is not in test_registered_usernames.json and email is not in test_registered_emails.json Both functions repeats until a unique field has been generated
unique email	If user attempts to input an email that is already in the mock database (test_registered_emails.json), an error message will be displayed that says 'Email is already in use'	Email: email from mock database(test_registered_emails.json) Username: (unique and random username not in test_registered_usernames.json)	Test passed	The email used will be the most recent email entered into the mock database (test_registered_emails.json) so that it tests different emails instead of only 1

		School year: 3 Password: test123! Confirm password: test123! ‘Register’ button pressed		
unique username	If user attempts to input a username that is already in the mock database (test_registered_usernames.json), an error message will be displayed that says 'Username is already in use’	Email: (unique and random email not in test_registered_emails.json) Username: Username from mock database (test_registered_usernames.json) School year: 3 Password: test123! Confirm password: test123! ‘Register’ button pressed	Test passed	The username used will be the most recent username entered into the mock database (test_registered_emails.json) so that it tests different emails instead of only 1
username length	If a user enters a username that isn’t between 4 and 12 characters long an error message will be displayed that says ‘Username must be 4-12 characters long’	Email: test@examplecom Username: abc School year: 3 Password: test123! Confirm password: test123! ‘Register’ button pressed Username: abcdefghijklm ‘Register’ button pressed	Test passed	2 instances performed 1 has 3 characters and the other has 13 characters
password	If a user enters a password that does not meet requirements an error message	Email: test@examplecom Username: abcd School year: 3	Test passed	5 instances performed: 1- password input is missing a piece of punctuation

	will be displayed that says ‘Password must be at least 8 characters long, include a number, an uppercase letter, a lowercase letter, and a symbol’	Password: Test123 Confirm password: Test123 ‘Register’ button pressed Password: test123! Confirm password: test123! ‘Register’ button pressed Password: Testtest! Confirm password: Testtest! ‘Register’ button pressed Password: TEST123! Confirm password: TEST123! ‘Register’ button pressed Password: Test12! Confirm password: Test12! ‘Register’ button pressed		2- password input is missing an uppercase letter 3- password input is missing a number 4- password input is missing a lowercase letter 5- password input is not 8 characters long
confirm password	If a user enters passwords that do not match an error message will be displayed that says ‘Passwords do not match’	Email: test@exampleco m Username: abcd School year: 3 Password: Test123! Confirm password: Test123? ‘Register’ button pressed	Test passed	-

school year	If a user enters a school year that isn't between 1 and 6 an error message will be displayed that says 'School Year must be between 1 and 6'	Email: test@exampleco m Username: abcd School year: 0 Password: Test123! Confirm password: Test123! 'Register' button pressed School year: 7 'Register' button pressed	Test passed	2 instances performed 1 has a school year of 0 and the other has a school year of 7
school year decimal	If a user enters a school year that isn't a whole number an error message will be displayed that says 'School Year must be a whole number'	Email: test@exampleco m Username: abcd School year: 4.1 Password: Test123! Confirm password: Test123! 'Register' button pressed	Test passed	
email	If a user enters an email that doesn't reach requirements an error message will be displayed that says 'Please enter a valid email address'	Email: test@exampleco m Username: abcd School year: 2 Password: Test123! Confirm password: Test123! 'Register' button pressed email: testexample.com 'Register' button pressed email: test@.com 'Register' button pressed email: @example.com	Test passed	6 instances of invalid inputs performed: 1- email is missing full stop 2- email is missing '@' 3- email is missing middle text 4- email is missing first text 5- email is missing last text 6- after full stop there is 1 character

		‘Register’ button pressed email: test@example. ‘Register’ button pressed email: test@example.c ‘Register’ button pressed		
Back	Back on register should navigate users to login	Built in web back button performed	Test passed	Back button removed from appbar so built in back function used

figure 6.6. Coded functional test plan and results: register page

Home/math club

Note: all tests that view the question page (home/maths club) and profile page require importing ‘fakeUser.dart’ as firebase fields are present. It has been imported and used in previous tests and will be used in tests moving forward. If the test is based on these pages or navigates to these pages fakeUser is required to mock firebase services.

Changes have been made to the question page to accommodate for testing:

Ability to support test users has been added as it gets injected data in oppose to firebase data in test cases.

Test users learnage is calculated separately to real users as it doesn’t require firebase values. Test users still add advanced mode to the school year to get learnage, they just use injected data. Learnage 8 and 9 are both unique to test users. Learnage 8 has a single set of a question, image and answers. Answers aren't random, only their position is. Learn age 9 still has a set question and correct answer but false answers are random within a range of the correct answer and are shuffled. Real users can’t access these learnages as the app won’t allow them to. To access these learnages for tests I can inject higher school years as data. These learnages are used in multiple tests.

To test logout I had to update the method to navigate test users to the login page, a method was also defined in fakeUser to mock signout that sets currentuser to null.

Renderflex problems occurred in tests but not in real use so I made the height of the main container null in test cases.

Question widget implements an addListener method in initState to listen for learnage changes, which triggers an update to generate questions based on new details.

All changes don't affect real cases or functionality

Test name	Tests	Inputs	Results	Notes
appbar	Banner is present with correct properties and contains text: ‘Math Club’	null	Test passed	-
Back	Back on question page should not navigate users	Built in web back button performed	Test passed	Back button removed from appbar so built in back function used
Navigation	Clicking the dropdown menu and then one of the options ‘about us’ ‘profile’ or ‘faq’ navigates users to the corresponding page and navigating ‘back’ from that page navigates users to the home page.	Drop down menu pressed ‘Profile’ button pressed Navigation back Drop down menu pressed ‘About Us’ button pressed Navigation back Drop down menu pressed ‘FAQ’ button pressed Navigation back	Test passed	-
logout	Clicking the dropdown menu and then ‘Logout’ option will dissociate users' information and send them to the login page.	Drop down menu pressed ‘Logout’ button pressed	Test passed	Logout was mocked for test users in both the question page and fakeUser file. When logout is pressed fakeUser sets user Id to null and question page navigates user to login page.
Container	Container is present with correct properties.	null	Test passed	-
Container contents	Question contents including, question, answers and image are present	null	Test passed	This question uses learnage 8 so question answers and images will not change.

	User data features like score out of 10 and days in a row are also present with injected data. Injected data is 3 for days in a row And 5 for score/10			Question, answers and images are called with learnage which is calculated with school year and advanced mode
Intro message	Check the presence of the question, image and answers. check answer feedback changes the colour of every answer button to green on answer to represent them all as correct.	First answer button pressed	Test passed	When users first register they are directed to the question page however their learnage hasn't yet been calculated as this is done when the question is called. Due to this I created an introduction question that says press any button to begin with every button being correct. To test this question I set learnage to 0 which is invalid so activates the else that holds this question.
Next question	First answer button pressed twice to call a new question. The image path, answer set, and question text are checked to be different to the first question.	First answer button pressed twice	Test passed	This test repeats until it passes (max 5 times but it won't need that many). This is because it is possible that the same image could be called twice as there are only 56 images. This feature uses getRandomQuestionAndAnswers(learnage) to call questions from questiongen.dart

timer	Next question should arrive after an answer button being pressed twice or after pressing an answer button once and waiting 3 seconds but not after 1 click as feedback is still showing .	Answer button pressed twice Answer button pressed	Test passed	Feedback isn't part of this test it is just the reason why question doesn't arrive after first click
incorrect feedback	After pressing the incorrect answer 'Incorrect' Text should appear and the correct answer should change colour to green. The selected answer should change red and other buttons shouldn't change colour.	Button containing text 12 pressed	Test passed	Learnage 8 used as answers are always the same so I know what colour they should change on answer
correct feedback	After pressing the correct answer 'Correct!' Text should appear and the correct answer should change colour to green. Other buttons shouldn't change colour.	Button containing text 10 pressed	Test passed	Learnage 8 used as answers are always the same so I know what colour they should change on answer
Answer position changes	Calling a new question by answering the previous one should call a new set of answers. These numbers should be the same answers as the question before but in a different order.	First answer button pressed 2 times Repeat if needed	Test passed	Learnage is set to 8 as the answer will not change but will be shuffled in the same way as actual answers. As answers are predetermined and their are only 3 possible positions, test can fail if numbers appear in the same place. Due to this test repeats until

				it passes.(max 10 times but it wouldn't need that many)
false answers	Calling a new question by answering the previous one should call a new set of false answers. These numbers should be different to the answers of the question before and not just in a different order.	First answer button pressed 2 times Repeat if needed	Test passed	Learnage is set to 9 as the correct answer wont change but false answers are generated the same as the other learnages within a range of the correct answer. As false answers are generated off the same correct answer for this question it repeats until it passes as although unlikely sometimes it could generate the same set twice as the range is only 7. (repeats max 10 times but it wouldn't need that many)
Learnage calculate	School year set to 1 and advanced mode set to true for test user. 25 Questions are added to a string and then a check is done that the string contains x and ÷ as these are values that appear only in learnage 2 up (I realised x can be found in a worded question from learnage 1 but this doesn't affect the test as both signs need	First answer button is pressed 50 times	Test passed	For test users, learnage is calculated in initstate and is static. Real users can change their learnage and it is recalculated after every question by recalculateLearnage. (however upon change it takes a couple of questions to update) For both real and test users learnage is defined with the same line of code. However

	to be found for test to pass)			for real cases advancedMode and schoolYear are called from firebase. Questions are called using learnage in the same way for both cases. Using getRandomQuestionAndAnswers(learnage)
score/10	correctCount set to 0 before the test. Button containing text 12 pressed twice then expected score/10 checked. After a correct answer when the score is at 10/10 it should no longer increase. The container that has the score should be red when score is 0, 1 and 2; orange when score is 3, 4, 5; yellow when score is 6, 7, 8, 9; and green if score is 10. Colour is checked before and after every score value that will change colour to check colour changes on the value it should and not before.	Answer button containing 12 pressed 22 times	Test passed	Updating firestore isn't performed for test users but for real users it is. Learnage 8 used as correct answer is always 12
Days in a row increase	When score/10 increases to 10, days in a row will increment by 1. In the test 'daysInRow' starts at 3 and 'correctCount' starts at 9. A correct answer will increase days	Answer button containing 12 pressed twice	Test passed	Updating firestore isn't performed for test users but for real users it is. Learnage 8 used as correct answer is always 12

	in a row to 4 and correct count to 10.			
Complete reset	If score out of 10 = 10 when reset is called days in a row wont reset and score out of 10 will.	Reset called	Test passed	Reset usually called at a certain time every day (00:00) but it can feature added so it can be called at any time Injected data: 'lastResetTimestamp' = DateTime.now(). correctCount = 10
Incomplete reset	If score out of 10 != 10 when reset is called days in a row and score out of 10 will reset.	Reset called	Test passed	Injected data: 'lastResetTimestamp' = DateTime.now(). correctCount = 9
Incomplete reset 2	If 'lastResetTimestamp' is more than a day before 'resetTimeToday' when reset is called days in a row and score out of 10 will reset.	Reset called	Test passed	Injected data: 'lastResetTimestamp' = DateTime.now().subtract(Duration(days: 2)) correctCount = 10

figure 6.7. Coded functional test plan and results: question/home page

Questiongen

Learnage tests can repeat until they pass (max 3 times but they won't need that), this is because each test gets a list of question texts and checks it for signs that appear in questions. It checks that every sign either does or doesn't appear at each learnage as required. Some signs like squared and cubed have a smaller probability of being selected for any given question in learnages that include them. Because of this sometimes not all signs would be found for a learnage. To resolve this these tests individually repeat until all expected signs are found.

Before I made the test repeatable, unexpected signs didn't show up where they shouldn't.

Worded questions are not tested here this just tests that the correct signs/question types come up or don't come up for specific learnages

Test name	Tests	Inputs	Results	Notes
Learnage 1	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 26 questions, Then the list is checked to make sure it contains '+' and '-' but doesn't contain '÷', '²', '³' or 'Square root of'	First answer pressed 52 times	Test passed	I removed a check that 'x' wasn't present as it is used in a worded question so may come up.
Learnage 2	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 26 questions, Then the list is checked to make sure it contains '+', '÷', 'x' and '-' but doesn't contain '²', '³' or 'Square root of'	First answer pressed 52 times	Test passed	
Learnage 3	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 26 questions, Then the list is checked to make sure it contains '+', '÷', 'x' and '-' but doesn't contain '²', '³' or 'Square root of'	First answer pressed 52 times	Test passed	
Learnage 4	Question text added to string of question texts then new	First answer pressed 52 times	Test passed	

	question called by pressing first answer twice. This repeats until there are 26 questions, Then the list is checked to make sure it contains '+', '÷', 'x' and '-' but doesn't contain '²', '³' or 'Square root of'			
Learnage 5	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 101 questions, Then the list is checked to make sure it contains '+', '÷', 'x', '-', '²', '³' and 'Square root of'	First answer pressed 202 times	Test passed	The following 3 tests are run with 101 questions in opposed to 26 as the '²', '³' and 'Square root of' have a lower chance of coming up, this reduces the likelihood that the tests will need to rerun.
Learnage 6	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 101 questions, Then the list is checked to make sure it contains '+', '÷', 'x', '-', '²', '³' and 'Square root of'	First answer pressed 202 times	Test passed	
Learnage 7	Question text added to string of question texts then new question called by pressing first answer twice. This repeats until there are 101 questions, Then the list is checked to make sure it contains '+', '÷',	First answer pressed 202 times	Test passed	

	'x', '-', '²', '³' and 'Square root of'			

figure 6.9.Coded functional test plan and results: profile page

Profile

My profile page wasn't test friendly after implementation as it was a closed class. Tests couldn't load the user details (majority of page) and when that was resolved tests could't access the snackbar. Also tests couldn't update values for advanced mode or details.

To resolve this I had to update the profile page to allow me to inject test data, and to mock output for firebase features that couldn't be accessed by tests. None of these changes affect the core functionality of the page or how real data is handled, these changes just affect how test data is handled.

Modified streams to optionally accept mock data for tests, with no impact on real user streams.

Added external ScaffoldMessengerKey support for snackbar output testing. This allows tests to access snackbar contents (messages to users).

Added the ability for test users to bypass firebase services for successful update and advanced mode to allow details or advanced mode to update in tests. To do this they use setState but bypass any firebase authentication or updates.

The UI layout is defined separately from the data handling logic, allowing testing of visual components and layout independently from Firebase services.

Additionally I have added the ability to change only 1 value from update details in contrast to only being able to change 2 fields together. This means that either of the 2 fields can be empty but not both. I have also added an error message for both fields being empty. This change applies to real and test users.

note
I manually tested learn age changing by running the app and changing it from the profile pages update details and toggle advanced mode, the difficulty changes on update but it takes a couple of questions. I couldn't test this for fake users as their learnage is calculated in init state.

Test name	Tests	Inputs	Results	Notes
appbar	Banner is present with correct properties and contains text: 'Math Club'	null	Test passed	-
loading indicator	When profile page is launched a loading indicator should be displayed	null	Test passed	-

Container	Container is present with correct properties.	null	Test passed	-
User details	Values displayed on profile (email, username, school year, advanced mode, days in a row, highest days in a row) are the same as the injected data	null	Test passed	-
highest day in a row update	If injected data includes daysInRow higher than highestDaysInRow then highestDaysInRow is set to daysInRow	null	Test passed	highestDaysInRow is checked against daysInRow during initialization and updated if lower. Firebase updates are bypassed in test cases.
Advanced mode	Pressing the advanced mode check box will put a tick in it and add a + 1 after school year value.	Press advanced mode check box	Test passed	setState applies to all users but firebase update is bypassed in test cases.
Forms	Forms appear with correct text, typable fields And close.	‘Update Profile’ button pressed Username: Test123! School Year Test123! ‘Cancel’ button pressed First ‘Change password’ button pressed Password : Test123! New password: Test123!	Test passed	‘Update Profile’ form and ‘Change Password’ form are both checked ‘Update Profile’ text finds 1 widget with ‘Update Profile’ text as the button to open form has spaces on both sides to make it the same size as Change Password button and the button on the form says update. Password form finds 3 widgets with text ‘Change Password’ as both buttons to

		Confirm new password: Test123! ‘Cancel’ button pressed		open the form and finalise update say Change Password.
Password changed	Opening ‘change password’ form, entering valid details to each field and pressing the last ‘change password’ button should update fake user’s password and the message ‘Password updated successfully’ should be displayed. A login attempt with the new password should then be successful	First ‘Change password’ button pressed Password : Test123! New password: newTest123! Confirm new password: newTest123! Last ‘Change password’ Login page built Email: 1234321@example.com Password: newTest123! ‘Login’ button pressed	Test passed	All messages related to updating password require scaffoldkey to get snackbar contents Password authentication is mocked in tests using PasswordCheckingFakeUser in fakeUser while real password update uses Firebase
Password requirements	Various invalid inputs are entered where the password doesn’t meet requirements. None should update the password, all should display an error message ‘New password must be at least 8 characters long, include a number, an uppercase letter, a lowercase letter, and a symbol'	First ‘Change password’ button pressed Password : Test123! New password: test123!! Confirm new password: test123! Last ‘Change password’	Test passed	Various cases are ran as there are multiple reasons this error message could display

		First ‘Change password’ button pressed Password : Test123! New password: Testtest! Confirm new password: Testtest! Last ‘Change password’ First ‘Change password’ button pressed Password : Test123! New password: TEST123! Confirm new password: TEST123! Last ‘Change password’ First ‘Change password’ button pressed Password : Test123! New password: Test1! Confirm new password: Test1!		

		Last ‘Change password’		
Passwords don't match	invalid input is entered where the passwords don't match. Test should not update password, test should display an error message ‘New passwords do not match'	First ‘Change password’ button pressed Password : Test123! New password: Test1234! Confirm new password: Test12345! Last ‘Change password’	Test passed	-
Incorrect password	Password entered doesn’t match injected test user password should display the error message ‘Incorrect current password’	First ‘Change password’ button pressed Password : Test1234! New password: newTest123! Confirm new password: newTest123! Last ‘Change password’	Test passed	-
New password same as old password	New password being the same as the old password should display error message: ‘New password must be different to current password’	First ‘Change password’ button pressed Password : Test123! New password: Test123! Confirm new password: Test123!	Test passed	-

		Last ‘Change password’		
details updated	Opening ‘Update Profile’ form, entering valid details to each field and pressing the ‘Update’ button should update fake users details on interface and message ‘Information updated successfully’ should be displayed.	‘Update Profile’ button pressed Username: (unique and random username not in test_registered_usernames.json) School year: 4 ‘Update‘ button pressed	Test passed	For this test firebase authentication is bypassed by changes on the profile page that only apply to test uses. For real users upon update firebase fields are updated but for test users it updates the local state All messages related to updating details require scaffoldkey to get snackbar contents
1 details updated	Opening ‘Update Profile’ form, entering valid details to one field and pressing the ‘Update’ button should update fake users details on interface and message ‘Information updated successfully’ should be displayed.	‘Update Profile’ ‘ button pressed School year: 4 ‘Update‘ button pressed ‘Update Profile’ ‘ button pressed Username: (unique and random username not in test_registered_usernames.json) ‘Update‘ button pressed	Test passed	-
No data	If the user doesn’t enter details and presses last ‘change	‘Update Profile’ ‘ button pressed	Test passed	fields can be empty so an error message is needed for

	password' button then error message 'Please enter at least one field to update' will be displayed	'Update' button pressed		everything being empty. On password this isn't needed as any field being empty prompts different error messages,
School year	If the user enter a school year value that isn't between 1 and 6 an error message 'School year must be between 1 and 6' will be displayed	'Update Profile' ' button pressed Username: test123 School year: 0 'Update' button pressed 'Update Profile' ' button pressed Username: test123 School year: 7 'Update' button pressed	Test passed	2 instances of invalid details tested: 0 and 7
School year decimal	If the user enter a school year value that isn't a whole number an error message 'School Year must be a whole number' will be displayed	'Update Profile' ' button pressed Username: test123 School year: 5.2 'Update' button pressed	Test passed	-
Username length	If the user enter a username value that isn't between 4 and 12 an error message 'Username must be between 4 and 12 characters long' will be displayed	'Update Profile' ' button pressed Username: abc School year: 3 'Update' button pressed 'Update Profile' ' button pressed	Test passed	2 instances of invalid details tested: abc and abcdefghijklm

		Username: abcdefghijklmn School year: 3 ‘Update‘ button pressed		
Username taken	If a username that matches one of the usernames in the mock database is entered when updating details an error message: ‘Username is already in use’ should be displayed.	‘Update Profile’ ‘ button pressed Username: user10000000 School year: 3 ‘Update‘ button pressed	Test passed	In the real test the input would be compared to the firebase database. However in this test it is compared to ‘test_registered_usernames.json’ in fake user
Back	Back on profile should navigate users to question page	Back icon pressed	Test passed	-

figure 6.9. Coded functional test plan and results: profile page

6.3. Coded non-functional test plan with results

Performance

The actual times of performance tests are recorded and used as output. Results vary between runs but here is an example of performance test output:

```
PS C:\Users\Harry Miller\Desktop\project\project> flutter test
00:16 +32: C:/Users/Harry Miller/Desktop/project/project/test/performance_
test.dart: Performance Tests Summary

Performance Summary:
Intro Page: 1092.00 ms
Login Page: 590.00 ms
Register Page: 191.00 ms
FAQ Page: 358.00 ms
About Page: 78.00 ms
Question Page: 170.00 ms
Profile Page: 98.00 ms
Login Process: 428.00 ms
Register Process: 598.00 ms
FAQ Search Response: 94.00 ms
FAQ Clear Response: 156.00 ms
Email Launch: 11.00 ms
Profile Update: 298.00 ms
Password Change: 196.00 ms
Advanced Mode: 47.00 ms
Question Generation: 213.00 ms
01:00 +82: All tests passed!
PS C:\Users\Harry Miller\Desktop\project\project>
```

figure 6.10.Screenshot of performance tests output

Test name	Test	Inputs	Results	Notes
Load pages	Pages should all load in under 2000ms	null	Test passed	Pages loaded while being timed 1 at a time.
Register	After entering details, From pressing 'register' button login process should take less than 2000ms	Email: (unique and random email not in test_registered_emails.json) Username: (unique and random username not in test_registered_usernames.json)	Test passed	After details have been entered, a timer is started and the 'register' button is pressed. When the home/question page is found the timer is stopped.

		School year: 3 Password: test123! Confirm password: test123! ‘Register’ button pressed		
Login	From pressing login button login process should take less than 1500ms	Email: 27946228@example.com Password: Test123! ‘Login’ button pressed	Test passed	After details have been entered, a timer is started and the ‘login’ button is pressed. When the home/question page is found the timer is stopped.
Questiongen	Question generation should take less than a second per question with stable internet connection.	First answer box pressed twice	Test passed	Timer starts before the button is pressed the first time. Timer is stopped after new text is found but before it is compared to old text to ensure change took place.
Change password	Changing password should take less than 1500ms	First ‘Change Password’ button pressed Current Password: Test123! New Password: NewTest123! Confirm New Password: NewTest123 Last ‘Change Password’ button pressed	Test passed	Before last ‘change Password’ button is pressed a timer is started and it is stopped when a confirmation message appears in the snackbar. Test passes if timer is stopped before 1500ms
Change details	Updating details should take less than 1500ms	‘Update Profile’ button pressed Username: (unique and random username not in test_registered_usernames.json)	Test passed	Before the ‘Update’ button is pressed a timer is started and it is stopped when a confirmation message appears in the snackbar. Test passes if

		School year: 4 ‘Update‘ button pressed		timer is stopped before 1500ms In a real environment firebase authentication may take slightly longer.
Advanced mode	Toggling advanced mode should takes less than 1500ms	‘Advanced mode’ checkbox pressed	Test passed	Timer starts before advanced mode is clicked and gets stopped after visual changes have taken place (box has a tick and + 1 next to school year value)
Launch email	Email launcher should launch in less than 1500ms	No input	Test passed	This doesn’t actually launch launcher it uses moclEmail so time recorded may not be accurate however on manual tests it launches within the 1500ms time frame.
Faq search	After searching results should update in less than 1500ms After clearing search results should update in less than 1500ms	Search: ‘q’ ‘Clear’ button pressed	Test passed	Timer started before q entered and stopped when results update Timer started before pressing clear and stopped when results update
Summary	Summarises performance test results, checks presence of all results before printing them	Other performance tests	Test passed Output printed	Combines all performance test results and prints output of: what is being tested and how long it takes to perform

figure 6.11. Coded non-functional test plan and results: performance

Compatibility
Manually tested app on edge and chrome. All features functioned as intended and with no inconsistencies.

Command: "cd build/web; python -m http.server 8080" needs to be run in a second terminal before "flutter test" is run for these tests to work properly. They need your web app running and accessible at http://localhost:8080 while testing. What makes these tests stand out is that they use actual browser instances in contrast to mocked environments.

I got drivers from files online:
https://storage.googleapis.com/chrome-for-testing-public/133.0.6943.94/win32/chromedriver-win32.zip
https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/?form=MA13LH
https://github.com/mozilla/geckodriver/releases

My file 'test_config.dart is used to run the drivers.

Note for tests to run you must have the version of the browser downloaded that matches the driver
For chrome: 135.x.xxxx.xx
For edge: 136.x.xxxx.xx
For firefox: 138.x.x(other versions of firefox may also work)

By default these tests are performed on windows 10.0.22631.4890 os.

Test name	Test	Inputs	Results	Notes
Chrome	All pages should load as browser instances using chrome driver	none	Test passed	Test uses ChromeDriver to verify all app pages load correctly in real Chrome browser. Driver version: ChromeDriver 133.0.6943.94
Edge	All pages should load as browser instances using microsoft edge driver	none	Test passed	Test uses MsEdgeDriver to verify all app pages load correctly in real microsoft edge browser. Driver version: msEdgeDriver 134.0.3124.66
Firefox	All pages should load as browser instances using gecko driver	none	Test passed	Test uses GeckoDriver to verify all app pages load correctly in real firefox browser. Driver version: GeckoDriver v0.34.0

figure 6.12. Coded non-functional test plan and results: compatibility

I would have liked to have tested safari on ios however I can’t do this without an apple computer so this isn’t feasible.

Firefox test crashed as a previous test wasn’t terminated, I fixed this by manually cancelling the process with the below command:
taskkill /F /IM geckodriver.exe

6.4. Manual grammar, accuracy, consistency and correctness check

To check this I ran the application and analysed everything and also checked the code.

Page	Issues
Intro	No issues
About us	-About us text- no punctuation in last sentence
Faq	-Faq "How do I answer questions?" answer- ‘what’ for ‘the’ -Faq "Is the app suitable for all children?" answer- remove ‘yes, ’ -Faq "How does score and days in a row work?" answer - swap ‘in’ for ‘from’, add ‘’
Profile	-Forms aren't consistently named -Update Profile form buttons aren't central --No error messages should have full stops -Password error message ‘a’ should be ‘an’ -Non consistent error message for unique username -make sure a decimal value isn’t valid for school year -password requirements error message disappears too quickly
Register	-No error messages should have full stops -Password error message ‘a’ should be ‘an’ -Non consistent error message for unique email and username -make sure a decimal value isn’t valid for school year -password requirements error message disappears too quickly
Login	No issue
Question	Takes a couple questions for learnage to update after details are changed or advanced mode toggled.

	After registering 1st question is blank and just says ‘other’ D for days in a row should be capital root not route for square root questions
Question gen	"Sam has 3 pens. Sam loses a pen. How many pens does Sam have left?"- incorrect answer "How many seconds in one minute?" add word “are” "How many hours in a day?" Switch “a” for “one” "What would we add to 56 to get 100?" Incorrect answer "What number is divisible by 3 and 6?" multiple correct answers "What's the product of 6 and 9?" Should be “What is” "What's the product of 7 and 8?" Should be “What is” "What is 2 sixths of 240?" Incorrect answer "How many sides on a Decagon? " should be “Does a decagon have?” not “on a Decagon?”’ "What is the mean of this data: 2, 3, 3, 5, 5, 6" answer not a whole number "What is 2 twelves of 72?" Incorrect answer

figure 6.13. Manual grammar, accuracy, consistency and correctness checks findings

6.5. Changes during and after testing

As mentioned in more detail under the functionality testing subheading and then the profile and home/math club subheading, changes have been made to the profile page and the home page during testing. Changes:
-Error message for both fields empty when updating details.
-ability to leave one of the two fields empty when updating details.
-profile page testability is improved.

-ability to support test user
-logout for test users
-test user learnage calculated separately to real users
-learnages 8 and 9 added for test users

New error message:

figure 6.14. Screenshot of new update details error message

As mentioned in more detail under the functionality testing subheading and then the registration subheading, I had to update the rules as they did not allow registration.

Changes from manual check

About Us
Comma added to the last sentence of about us paragraph.

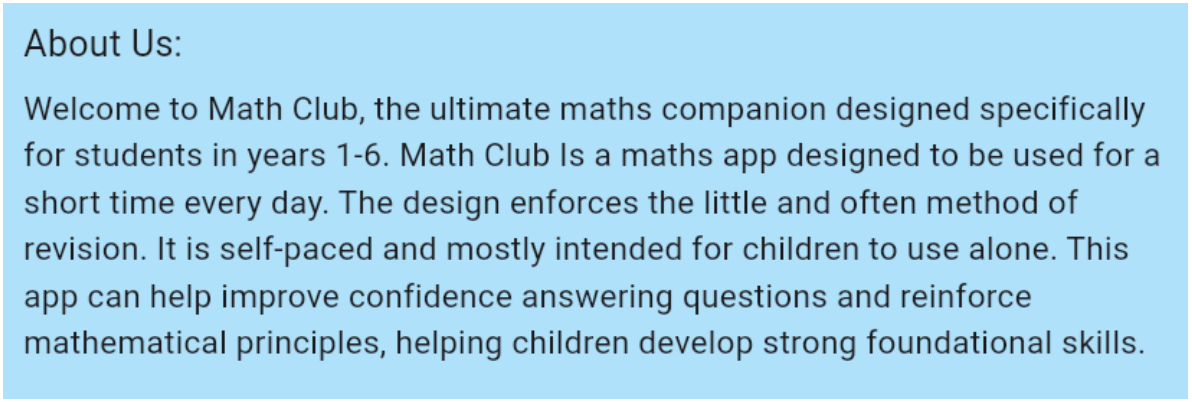


figure 6.15. Screenshot of about us paragraph

Faq
3 faqs had changes. 2 were for grammatical reasons and the other was to enforce correctness.

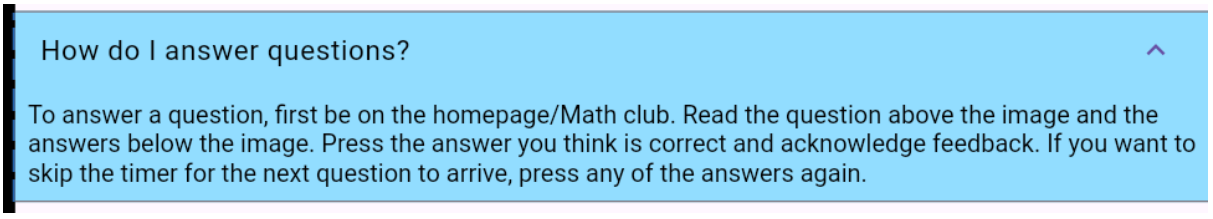


figure 6.16. Screenshot of updated faq 1

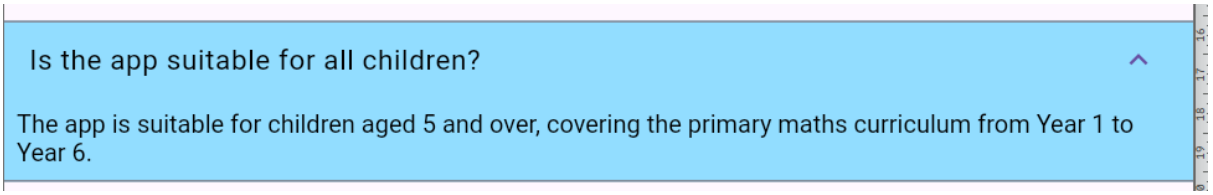


figure 6.17. Screenshot of updated faq 2

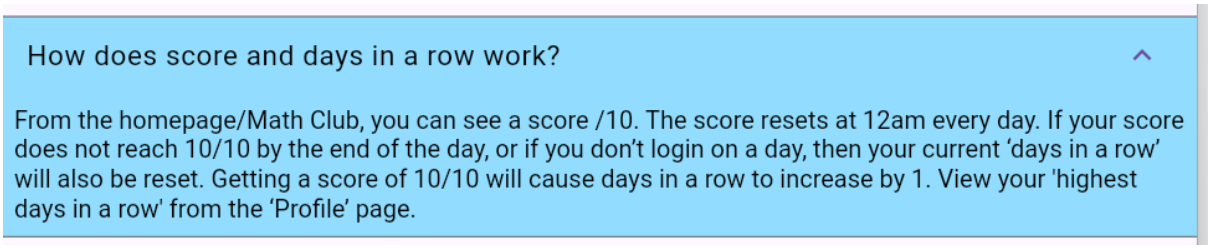


figure 6.18. Screenshot of updated faq 3

Register
A lot of snackbar messages from implementation ended in full stops in the register and profile page. I removed all of these fullstops to enforce consistency. I am not going to add updated screenshots as there were a lot of messages with fullstops and it is a small change.

Username uniqueness error message was changed to have consistency with other uniqueness error messages.

figure 6.19. Screenshot of updated username error message on register page

For password error message 'a' changed to 'an' to be grammatically correct. This error message is also displayed for 6 seconds in opposed to 3 seconds for the other error messages on this page.

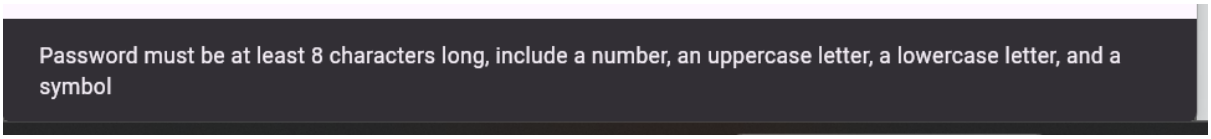


figure 6.20. Screenshot of updated password error message on register page

error message added that checks school year input is a whole number

figure 6.21. Screenshot of new school year error message on register page

Profile
Username uniqueness error message was changed to have consistency with other uniqueness error messages.

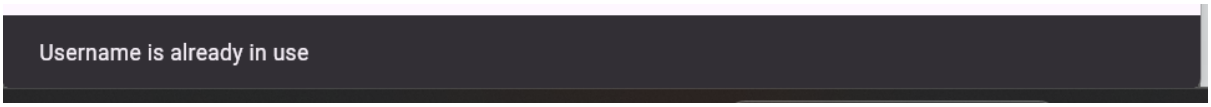


figure 6.22. Screenshot of updated username error message on profile page

For password error message 'a' changed to 'an' to be grammatically correct. This error message is also displayed for 6 seconds in opposed to 3 seconds for the other error messages on this page.

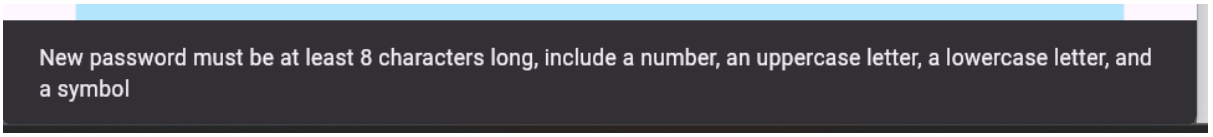


figure 6.23. Screenshot of updated password error message on profile page

error message added that checks school year input is a whole number

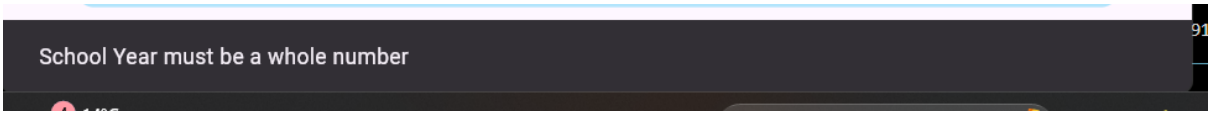


figure 6.24. Screenshot of new school year error message on profile page

Update Profile form title changed to match with the text in the button that opens it. This enforces consistency as the Change Password form title matches the text in the button that opens it. Buttons are also centralised.

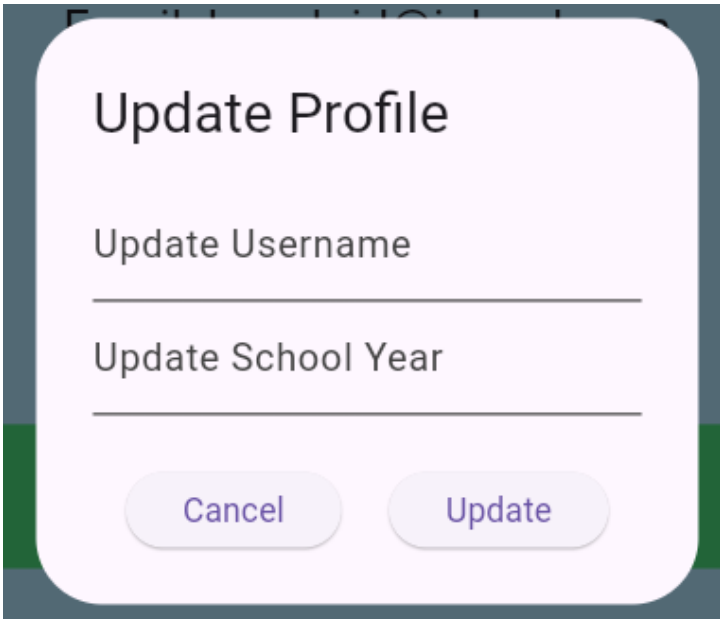


figure 6.25. Screenshot of updated update profile form

Question
Intro question added with every answer being correct

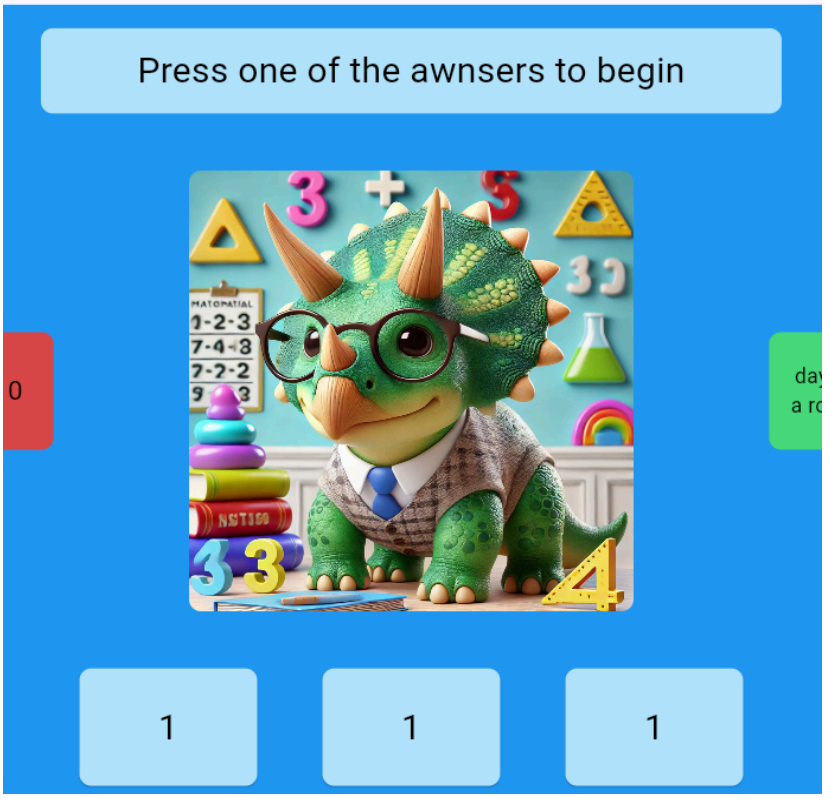


figure 6.26. Screenshot of new intro question

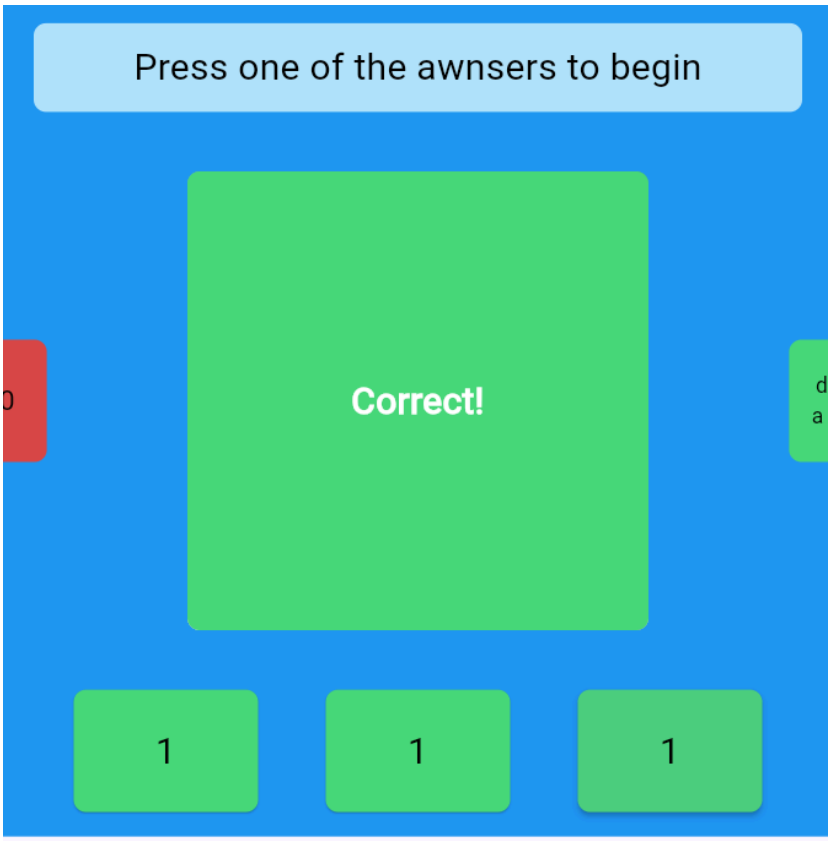


figure 6.27. Screenshot of intro question after answer

Capital D for days on question page

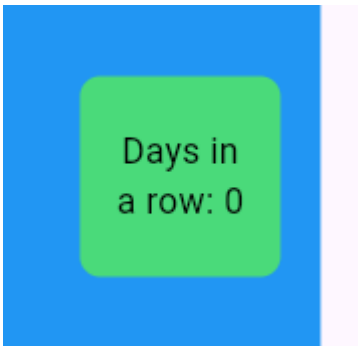


figure 6.28. Screenshot of updated days in a row display

Spelling of root corrected for learnages 5, 6 and 7

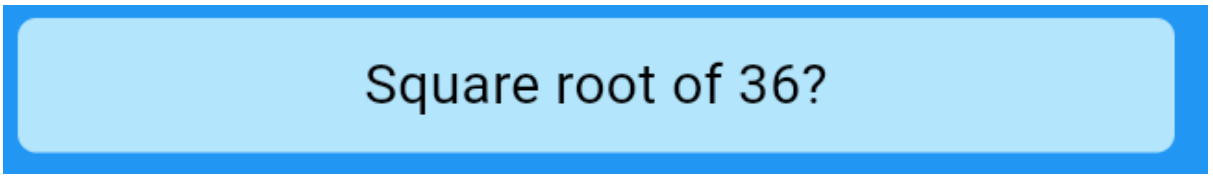


figure 6.29. Screenshot of updated square root question

Questiongen

Written questions and answers I changed:

Learnage 1, case 9
Sam has 3 pens. Sam loses a pen. How many pens does Sam have left?
2

Learnage 2, case 11
How many seconds are in one minute?
60

Learnage 2, case 12
How many hours are in one day?
24

Learnage 4, case 0
What would we add to 56 to get 100?
44

Learnage 4, case 18
What number is divisible by 4 and 12?
24

Learnage 4, case 23
What is the product of 6 and 9?
54

Learnage 5, case 8
What is the product of 7 and 8?
56

Learnage 5, case 22
What is 4 sixths of 240?
160

Learnage 6, case 7
How many sides does a decagon have?
10

Learnage 6, case 21
What is 2 twelves of 72?
12

Learnage 7, case 7
How many sides does a decagon have?
10

Learnage 7, case 21
What is 2 twelves of 72?
12

Testing

Functionality tests and their documentation above have been updated to reflect changes.

Readme added (see Appendix F: Readme) to explain how to run and use apps as well as how to run tests.

6.6. Usability study

Method of Analysis
I chose to carry out a cognitive walkthrough covering all core mechanics of my application. This involves creating personas and attempting to evaluate how their experience would be performing specific tasks.

Limitations
Limitations of my usability study include the fact that I performed it myself although trying to think from the perspective of my personas. This is due to time being a limitation of my overall project. A usability study with real users in my target audience would be necessary before deployment.

6.6.1. Personas

TARGET PERSONA IMAGE	PERSONA INFORMATION
	PERSONA NAME
	Billy Jones
	Location
	Portsmouth UK
	AGE
	5
	Education
	Attending Primary School

GOALS AND MOTIVATIONS
Billy wants to be able to use the website to improve his maths skills.

VALUES AND WORK METHODS
Billy loves being at school and mixing with his peers. Billy believes he is good at counting but does not like spelling or reading.

CHARACTERISTICS THAT APPLY TO THE APP
Billy is 5 years old and the app will help him keep up with the national curriculum requirements for a year 1 student.

figure 6.30.First persona: Billy Jones

TARGET PERSONA IMAGE	PERSONA INFORMATION
	PERSONA NAME
	Sam Clark
	Location
	Portsmouth UK
	AGE
	9
	Education
	Attending Primary School

GOALS AND MOTIVATIONS
Sam’s parents want him to improve his maths skills as he is slightly behind some of his peers

VALUES AND WORK METHODS
Sam enjoys being at school. He prefers playing with his friends and isn’t currently interested in anything academic.

CHARACTERISTICS THAT APPLY TO THE APP
Sam is 9 years old and the app will help him keep up with the national curriculum requirements for a year 5 student.

figure 6.31.Second persona: Sam Clark

6.6.2. Cognitive walkthrough

The first 6 tasks will start from the question/home page

Task 1
Can users find information about how days in a row works

Action Step: Navigate to the FAQ page by pressing ‘menu’ and ‘FAQ’.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☐	Billy may go to the about us page first. If he does I believe he would quickly realise he is in the wrong place, go back and perform the step correctly.
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Whether it is the first page they go to or not, I believe if users visit the other pages they will realise they are in the wrong place, and if they visit the faq page they will realise they are in the right place.		

Action Step: Scroll down to find "How does score and days in a row work?", optionally start typing ‘days in a row’ to filter options and find the question faster.		
	Yes	No - Why?
Was the action step successful?	☐	Billy may not be able to find it as there are a lot of questions about different things.
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☐	Billy may not be able to find the specific question he is looking for.

Action Step: Scroll down to find "How does score and days in a row work?", optionally start typing ‘days in a row’ to filter options and find the question faster.		
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Considering my young target audience some users may need help finding information, likely from a guardian or teacher. This is because of limitations my target audience can face such as reading ability.		

Action Step: Click question to display answer.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Considering users find the question they are looking for I believe they will be able to perform this step.		

Task 2
Can users find the email to contact the people behind the app

Action Step: Navigate to the About Us page by pressing ‘menu’ and ‘About Us’.		
	Yes	No - Why?
Was the action step successful?	☑	

Will the user try to achieve the right result?	☐	Billy may go to the faq page first. If he does I believe he would quickly realise he wes in the wrong place, go back and perform the step correctly.
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Whether it is the first page they go to or not, I believe if users visit the other pages they will realise they are in the wrong place, and if they visit the faq page they will realise they are in the right place. Note that this task isn't usually something children would do as it is more for guardians and teachers. Regardless, my target audience is children and it is a feature my app offers, therefore I will still view the task from the perspective of my personas in my target audience.		

Action Step: Find email after 'Email' text, optionally press the linked email to contact it.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Considering users understand what an email is, I think they will be able to find it when on the about us page.		

Task 3
Can users answer questions

Action Step: Calculate answer to question		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I think users will be able to see and attempt the questions, whether the answer they come up with is correct or not.		

Action Step: Press the answer they think is correct, optionally press any answer again to skip timer and feedback.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	

Comments: If the answer users calculated isn't present, they would have got it wrong, this should prompt users into attempting to recalculate their answer before continuing. They may not initially realise they can skip feedback but this shouldn't have a big impact on their experience.

Task 4
Can users change their password

Action Step: Navigate to the profile page by pressing 'menu' and 'Profile'.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would be able to perform this step.		

Action Step: press 'Change Password' button, fill in fields, press 'Change Password' button		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see	☑	

that progress is being made towards the goal?		
Comments: I believe users will be able to attempt this form even if they don’t input valid details.		

Action Step: Until details update and success message is shown, redo form in response to the last error message		
	Yes	No - Why?
Was the action step successful?	☐	Form closes after invalid input so both personas may struggle to keep track of the information they are trying to enter and error messages. I believe if Sam persists he will be able to change his password but Billy may not.
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: The forms closing may be frustrating for users and slow down the process of changing passwords. It is an issue that needs resolving before deployment.. However for users that can't change their password it may be better for them to get assistance from an adult to reduce the likelihood of them forgetting their password and getting locked out of their account.		

Task 5
Can users change their details

Action Step: Navigate to the profile page by pressing ‘menu’ and ‘Profile’.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would be able to perform this step.		

Action Step: press ‘Update Details’ button, fill in fields, press ‘update’ button		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users will be able to attempt this form even if they don’t input valid details.		

Action Step: Until details update and success message is shown, redo form in response to the last error message		
	Yes	No - Why?
Was the action step successful?	☐	Form closes after invalid input so both personas may struggle to keep track of the information they are trying to enter and error messages. I believe if Sam persists he will be able to change his details but Billy may not.
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: The forms closing may be frustrating for users and slow down the process of changing details. It is an issue that needs resolving before deployment.. However for users that can't update their details it may be better for them to get assistance from an adult to reduce the likelihood of valid but incorrect details being entered.		

Task 6
Can users turn on advanced mode

Action Step: Navigate to the profile page by pressing ‘menu’ and ‘Profile’.		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	

After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would be able to perform this step.		

Action Step: Press ‘Advanced Mode’ checkbox		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would be able to perform this step		

The last 2 tasks will start from the introduction page

Task 7
Can users register

Action Step: press container		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	

Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Comments: I believe users would all be able to complete this step as the container is clearly labelled telling users to click it..		

Action Step: press ‘Don't have an account? Register’ button		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would all be able to complete this step as the button is clearly labelled.		

Action Step: fill in form fields and press ‘Register’ button		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	

Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users will be able to attempt this form even if they don't input valid details.		

Action Step: Respond to error messages by updating fields and press the 'Register' button. Repeat this step until directed to question page		
	Yes	No - Why?
Was the action step successful?	☐	Creating an account includes: entering an email, entering an uppercase letter and sign for password along with other fields. Billy may not succeed due to his technical ability with computers and literacy skills.
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: Clear error messages that describe specific issues with the users input allow them to respond to the issues. Input remains present after unsuccessful register attempts so users can edit it until no issues persist. All required fields are completely necessary and non negotiable. For users that lack the technical ability to create an account by themselves, it is better that they have help from an adult. This significantly decreases the likelihood of users entering incorrect information that could negatively impact their experience.		

Task 8
Can users login

Action Step: press container		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users would all be able to complete this step as the container is clearly labelled telling users to click it..		

Action Step: fill in form fields and press ‘Login’ button		
	Yes	No - Why?
Was the action step successful?	☑	
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: I believe users will be able to attempt this form even if they don’t input valid details.		

Action Step: Respond to error messages by updating fields and press the ‘Login’ button. Repeat this step until directed to question page		
	Yes	No - Why?
Was the action step successful?	☐	If users don’t know their details they won’t be able to gain access to their account.
Will the user try to achieve the right result?	☑	
Will the user notice that the correct action is available?	☑	
Will the user associate the correct action with the effect they are trying to achieve?	☑	
After the action is performed, will the user see that progress is being made towards the goal?	☑	
Comments: This could be resolved by having multiple error messages for login and adding a forgot password feature.		

figure 6.32. Cognitive walkthrough

Overall this study shows, the application is highly usable by the target audience with some minor limitations. The youngest users may need adult assistance with account management and form completion tasks, but this is to be expected. The biggest concerns are lack of support for users who forgot their password, and profile page forms closing on unsuccessful attempts. The majority of features demonstrated no significant usability issues.

6.6.3. Fault tolerance

I have analyzed my app for actions users can make that can potentially have negative consequences. I will analyse these instances to evaluate my apps fault tolerance. There may be some overlap with the findings of my cognitive walkthrough.

Invalid entry register	Register has multiple error messages that are displayed for invalid input, each field has an error message and some fields have 2 error messages describing different issues. The error messages clearly explain to the user the issue with their input. Furthermore if input is invalid fields remain filled in, this means users can respond to the error message without having to enter in all the fields again. A limitation is that only one error message can be displayed at a time, so if 2 fields or more have issues users will have to resolve the first error message before the next can be displayed. This doesn't pose a major problem and the register process is ok and usable as it is.
Invalid entry login/details forgotten	If users have forgotten or don't know their details there is a lack of support. There is no forgotten password feature and only one possible error message. I believe implementing a forgotten password feature and 2 error messages for the email not being registered and the password being incorrect would improve fault tolerance and error recovery. These changes give users insight to the issue with their input and if their email is registered they would be able to change their password and regain access to their account.
Invalid entry change password and update details This applies to both forms on profile page (update details and change password) as there are only small differences between them.	Similar to the register page the forms have multiple error messages for invalid input describing the specific nature of the issue. An issue I found with the forms is that they close on invalid input. This means if the user enters invalid input to either form they will need to fill in all the fields again. Also only one error message is shown each time the user fills in one of the forms, so if there are multiple issues the user would have to fill in the form once for each of them until there are no issues. It could be difficult for users to keep track of the error messages as well as the details they plan to enter. This could be resolved by keeping the forms open until the user closes it or successfully fills it in.
Accidentally change school year	If a user changed their school year to the wrong year without realising and returned to answering questions they may not realise until they return to the profile page. A solution to this could be adding the users learnage to the question page so they can see if it is incorrect while answering questions.

Accidentally logs out	It is possible that users could accidentally press the logout button, this would log them out and cause a minor inconvenience. However the logout button is in the dropdown menu so to press it users would first have to open the menu, making it harder to do by mistake. The buttons all contain large, clear text to make them easier to read. I believe I have done as much as I can do to prevent this mistake. I don't think it is likely to happen and if it does i don't think it would be down to faults in my app.
Days in a row resets	Days in a row resetting when users don't achieve a 10/10 score by the end of any day is a core functionality of my application. As much as it is a negative thing that users shouldn't want to happen, if it didn't reset there would be less incentive to complete score/10 on a daily basis. Therefore this function doesn't pose any problems and should stay as it is.

figure 6.33. Fault tolerance analysis

6.6.4. Summary/changes

This usability study was performed late in testing, so due to the constraint of time there won't be any more implementation or testing changes in this project. The changes that need to be made due to this study will be noted and completed before deployment. Things that need to be changed based on this study are:

-Learnage on question page
-Profile page forms closing
-Forgot password(evaluating usability did resurface the idea of implementing a forgot password feature although I already knew i wanted to add it)
-Multiple login error messages(evaluating usability did resurface the idea of implementing multiple login error messages although I already knew i wanted to add it)

These changes to make are explained in more detail in the evaluation chapter under the heading 'What's left', along with other ideas that didn't come from this usability study.

6.7. Justifying testing choices

I will not evaluate how I reached my functional and non-functional requirements until the evaluation chapter, this is just to explain my thought process behind my decisions.

Functional tests

Firstly I tested all functional features of my app. This involved systematically verifying that each component worked as expected according to the functional requirements and implemented features. I performed these tests as it is vital to ensure the app functions as expected, furthermore it is a requirement of this project. I would have liked to test the learning age being re-calculated however for test users it is calculated in init state. I did however manually test updating the learn age from profile page and checking difficulty changes.

Non-functional tests

Performance

I tested various load times to avoid the event of slow performance frustrating children and reducing engagement.
User load testing would need to be performed before deploy but due to time constraints it could not be included

Compatibility

For compatibility I tested 3 browsers on windows. I would have liked to have tested safari on ios however due to the limitation of me not having access to an apple device this was not an option. This is something that would need to be tested before deployment to ensure accessibility.

Manual grammar check

I am aware I may have made grammatical or correctness mistakes during implementation. For this reason, I performed a manual check of the entire app in search of mistakes. This search found more mistakes than I was expecting but resolving these mistakes improved my project.

Usability

Usability is a crucial factor when developing an app for school children. The usability directly impacts how effectively they can learn and engage with the content. Children are less efficient at using technology than adults, so it is vital that the app caters to their usability needs.

Due to the limitation of time I have been unable to perform a usability study on real users in my target audience. This is something that I need to do before deployment but due to the limitation of time, this wasn't able to be included in this project. This study would require research on laws regarding performing a research study on children. This study would include general usability testing and difficulty testing.

Security

Due to constraints (time and the knowledge I have learnt on my degree), I chose against testing security. The app is currently a mock, however if this app were to be deployed security testing would be necessary.

I was previously concerned about my rules as they don't have restrictions, however only I have the firebase api key. Therefore only I have database access and users can only change their own information from within the app.

Chapter 7: Methodology

7.1. Development methodology

This study employed a mixed methods approach combining both qualitative and quantitative research methods to develop, implement, and test the Math Club application. The development process was structured into four sequential sections.

7.1.1 Literature Review/research

Research included analysing mathematics education research, usability research, evaluating existing math applications and reviewing the UK National Curriculum. The findings shaped both functional and non-functional requirements, focusing on usability and improving learning effectiveness. The literature review was used to determine the feasibility of my project and research was used to inspire following chapters.

7.1.2. Design Methodology

The design process followed a progressive, child-focused methodology that prioritised the needs and capabilities of primary school users at each development stage. Creating initial concepts focused on simplicity.

- Developing architecture designs with written plans and diagrams made with lucid app.
- Creating multiple generations of ui design with lucid app and hand drawn
- Choosing requirements mostly based on design.

7.1.3. Implementation methodology

Development followed a component-based approach:

Frontend:

- One by one building of ui's with their functional features programmed using flutter/dart on vs code.

Backend:

- Firebase: For authentication and data storage
- Database: Created database.dart to define the database and its fields and support firebase.
- Question Algorithm: developed algorithm to generate random, age appropriate, curriculum aligned questions.
- Other back end code: Some functional features required backend code(mostly firebase features). This can be found in auth.dart
- Rules: Adjusted firebase rules multiple times for security and/or functionality purposes

7.1.4. Testing Methodology

Comprehensive testing validated the application:

- Functional Testing: 92 automated tests verifying features programmed using flutter/dart on vs code.
- Non-Functional Testing: Performance and browser compatibility testing programmed using flutter/dart on vs code.
- Usability Testing: Persona development and cognitive walkthroughs.
- Manual check: correctness and grammar check of running app and code.
- Responding to tests: Some tests required implementation changes to be resolved(changing original code). This wasn't planned but was necessary after performing tests.

7.2. Other methodologies

Data Analysis

Analysis included both quantitative measures (test pass rates, performance metrics) and qualitative assessment (design improvements, usability evaluation).

Ethical Considerations

The project adhered to guidelines for child-focused applications, including data protection, age-appropriate content, and accessibility considerations.

Supervisor Consultation

Regular meetings with my project supervisor provided guidance and feedback throughout my development journey.

Risk mitigation

I successfully stayed ahead of where I thought I needed to be throughout the project. This allowed me to have more time for bug fixes and testing which was retrospectively needed.

I strategically broke my project into sections and then manageable subsections. This allowed me to make more significant progress without becoming overwhelmed.

Right up

I documented some of my right up as I went along and other parts later after they had already been completed. This led to confusion and things being more complicated than they had to be. If I do a right up like this again in the future I would right up as I go along.

Although my organisation has affected me while completing the report I don't believe it to have had a negative effect on the report itself

Chapter 8: evaluation

8.1. Original functional requirements and where they were met

Original requirements	How they were met	Test reference
Backend database	_	_
Firebase database Stores all data required for users Read/writes data when needed for app functionality	App integrates a firebase database for storing, reading and writing user details as part of various processes(login, register, update details, change password, user details, question generation and more) across various pages(login, register, profile, home). The backend database functions as intended in all cases therefore meets requirements.	Manually tested
User authentication control	_	_
User must be able to log in Successfully logging in sends user to homepage	User login implemented on login page and successful login tested by 'Successful login' in login_test.dart	-'Successful login' in login_test.dart
System must take input from the user: username and password.	User input required for successful login. Fields for login are tested by 'Typable fields' in login_test.dart. No input for login is tested as part of 'Incorrect login' in login_test.dart.	-'Typable fields' in login_test.dart -'Incorrect login' in login_test.dart
System must validate the input using firebase	In the case of real users firebase is integrated into login and on successful login their stored details and progress are displayed. This has been manually tested test users login is mocked in fakeUser.dart and doesn't require access to firebase	Manually tested
Error messages must be present when the user enters invalid input.	An error message has been implemented on the login page for when login is unsuccessful.	-'Incorrect login' on login_test.dart

	This message is tested as part of ‘Incorrect login’ on login_test.dart. The test also makes sure users remain on the login page.	
User account creation control	**-**	**-**
Users can create accounts Successfully registering sends user to homepage	This has been implemented on register page and tested by ‘Register’ in register_test.dart	-‘Register’ in register_test.dart
System must take user input: username, school year and password.	User input is required to register. Fields are tested by ‘Typable fields’ in register_test.dart	-‘Typable fields’ in register_test.dart
System creates an account if input is valid and stores users input in firebase database.	This has been implemented and manually tested. For test cases, register doesn’t use or update firebase as it is mocked in fakeUser.dart.	-
Usernames must be 4 characters and unique.	2 checks with error messages ensure this. 1 check is to make sure the username is between 4-12 characters long, and the other is to make sure the username is unique. These are tested respectively by ‘username length’ and ‘unique username’ in register_test.dart.	-‘username length’ in register_test.dart -‘unique username’ in register_test.dart
Passwords must be 8 characters long and include at least 1 number and symbol	Check has been implemented to make sure the password is at least 8 characters long, include a number, an uppercase letter, a lowercase letter, and a symbol. Error message is also present. Various cases of invalid passwords tested by ‘password’ in register_test.dart.	-‘password’ in register_test.dart
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.	There are multiple error messages with respective text describing the nature of the issue with users input. Each error message has its own test, some tests have various cases	‘unique email’, ‘school year’, ‘school year decimal’, ‘username length’, ‘password’, ‘confirm password’,

	of invalid input that triggers the same error message. Error message tests are ‘unique email’, ‘school year’, ‘school year decimal’, ‘username length’, ‘password’, ‘confirm password’, ‘email’ and ‘unique username’ in register_test.dart.	‘email’, ‘unique username’ in register_test.dart.
User profile	-	-
Users can read a selection of their details stored in firebase. These details may be: days in a row, highest days in a row, username and school year.	In test cases data details are injected not from firebase, but there presence and ui are tested by ‘User details’ in profile_test.dart	-‘User details’ in profile_test.dart
Users can toggle advanced mode from this page updating their question difficulty and firebase.	Advanced mode has been implemented and tested. The toggle for advanced mode is on the profile page. This is mentioned in more detail under functional requirement: advanced mode below.	SEE ADVANCED MODE REQUIREMENT BELLOW
Question generation	-	-
Users must be able to answer questions.	This has been implemented. It doesn’t have its own test but answering questions is part of multiple tests including: ‘Next question’, ‘correct feedback’, ‘incorrect feedback’ and ‘timer’ in question_test.dart.	‘Next question’, ‘correct feedback’, ‘incorrect feedback’ ‘timer’ in question_test.dart.
Difficulty levels must be accurate to corresponding school years.	Question types,/topics and number ranges where researched based of the national curriculum for their respective school year	5.4. Research / Appendix E: Text directly from national curriculum with chosen values
Randomly generating questions with answers at a difficulty based on the user's school year and advanced mode.	This has been implemented and is tested by ‘Learnage calculate’ and ‘Next question’ in question_test.dart. There is a test for all 7 learnages in questiongen_test.dart. These tests ensure each learnage is generating questions of the correct question type.	-‘Learnage calculate’ in question_test.dart -‘Next question’ in question_test.dart All of questiongen_test.dart

Randomly generating false answers and position of correct answer.	This has been implemented and is tested by ‘false answers’ and ‘Answer position changes’ in question_test.dart.	-‘false answers’ in question_test.dart -‘Answer position changes’ in question_test.dart
Questions come with the question itself, an image and 3 options as answers.	This has been implemented and is tested by ‘Next question’ in question_test.dart. This test also ensures the new question, answers and image are different to previous.	-‘Next question’ in question_test.dart
Answer feedback present after selection.	This has been implemented and is tested by ‘correct feedback’, ‘incorrect feedback’ and ‘timer’ in question_test.dart.	-‘correct feedback’, in question_test.dart. -‘incorrect feedback’ in question_test.dart. -‘timer’ in question_test.dart.
Daily score	_	_
Score will increase with every correct answer until it has reached 10 where it will remain until reset at 12am every day.	This has been implemented and tested. Score increasing with correct answers and being capped at 10 is tested by ‘score/10’ in question_test.dart. I decided that I want the box to change colour as users score increases. This has been implemented and the same test checks colour change occurs when it should. Reset has been implemented and tested by 'reset incomplete', 'reset incomplete 2' and 'reset complete' on question_test.dart.	-‘score/10’ on question_test.dart - 'reset incomplete' on question_test.dart -'reset incomplete 2' on question_test.dart. -'reset complete' on question_test.dart
Days in a row	_	_
Days in a row stores users current streak of completing 10 questions on consecutive days.	Days in a row is a firebase field it can be stored read and updated when necessary	-
Increases by 1 when users achieve 10/10 on any day.	This has been implemented and tested by 'days in a row increase' on question_test.dart.	-'days in a row increase' on question_test.dart
Resets with score if they don’t achieve 10 on any given day.	This has been implemented and tested by 'reset	- 'reset incomplete’ on question_test.dart

	incomplete', 'reset incomplete 2' and 'reset complete' on question_test.dart.	-'reset incomplete 2' on question_test.dart -'reset complete' on question_test.dart
Highest days in a row would also be stored and updated with days in a row when necessary.	This has been implemented and tested by 'highest days in a row update' on profile_test.dart.	-'highest days in a row update' on profile_test.dart
Advanced mode	_	_
User has the ability to toggle advanced mode. toggling advanced mode shows visual feedback	If the advanced mode checkbox is pressed it is filled in with a tick and a '+ 1' appears next to the school year. This is tested by 'Advanced mode' in profile_test.dart	-'Advanced mode' in profile_test.dart
toggling advanced mode adds 1 to users' school year so questions called will be a higher difficulty.	For real users, updating advanced mode updates its field in firebase. This is then used to recalculate learnage For test users, learnage is calculated in init state but does use school year + advanced mode like real users. This means learnage can't be updated in tests without injecting new data and rebuilding state. I did test that the calculated learnage for test users is school year and +1 if advanced mode is true in 'Learnage calculate' in question_test.dart and manually tested for real cases.	-'Learnage calculate' in question_test.dart
For year 6 users in advanced mode a new difficulty level will be made.	7 learnages have been made to accommodate for users in year 6 using advanced mode. Each has been tested to make sure it includes the correct signs in questiongen_test.dart.	-'Learnage 7' in questiongen_test.dart /all of questiongen_test.dart

	These tests injected data includes advanced mode: true and false at different learnages so it's tested further.	
Faq page	-	-
Users must be able to find solutions to issues they may have in faq. Questions must help users understand the app and explain all its features.	Faq page explains all features of the app and how they work.	-'faqs' in faq_test.dart
Faq page must be searchable.	Faq search has been implemented and has a clear button, this is tested by 'Search' in faq_test.dart	-'Search' in faq_test.dart
Option to click on questions to view answers.	Qas are clickable; this is tested by 'faqs' in faq_test.dart, along with the presence of every question and answer.	-'faqs' in faq_test.dart
Most likely scrollable but depends how many questions.	This page is scrollable and scrolling is required for 'Search' and 'faq' in faq_test.dart	-'Search' in faq_test.dart -'faq' in faq_test.dart
About us page	-	-
Contact information including email, postal address and phone number must be present. Information about the app must be present.	All relevant information is present. Its presence is tested by 'text', 'Email', 'Phone Number' and 'Postal Address' in about_test.dart.	-'text', 'Email', 'Phone Number', 'Postal Address' in about_test.dart
Email address should have clickable link	Email link was implemented and tested by 'Email' in about_test.dart.	-'Email' in about_test.dart
Navigation	-	-
Users must be able to easily navigate between pages. App must offer simple navigation.	There are multiple ways to navigate that are designed, implemented to be simple and easy to use. The login and registration pages both have a clear button to get to the other. This is tested by 'Register' on	-'Register' on login_test.dart -'login' on register_test.dart -'container' in intro_test.dart -'Successful login' on login_test.dart

	login_test.dart and ‘login’ on register_test.dart. The introduction page offers clear navigation to continue. This is tested by ‘container’ in intro_test.dart. Logging in or registering automatically directs users to the question page. This is tested by ‘Successful login’ on login_test.dart and ‘Register’ on register_test.dart. The question page offers a dropdown menu with options About us, Faq and Profile that directs users to respective pages; and Logout that logs users out and directs them to login.This is tested by ‘Navigation’ and ‘logout’ on question_test.dart. About us, Faq and Profile offer back buttons to the question page. Back is tested on all pages, even those that don’t have a button. Some pages have back completely disabled such as question page and intro. This is tested by ‘Back’ on about_test.dart, faq_test.dart, intro_test.dart, login_test.dart, profile_test.dart, question_test.dart and register_test.dart Faq offers links to other pages. This is tested by ‘Links’ in faq_test.dart.	-’Register’ on register_test.dart -‘Navigation’, ‘logout’ on question_test.dart -‘Back’ on about_test.dart, faq_test.dart, intro_test.dart, login_test.dart, profile_test.dart, question_test.dart and register_test.dart -‘Links’ in faq_test.dart
Change password	**_**	**_**
Users must be able to change their password.	This feature has been implemented. To change their password, users have to open a form, enter: current password, new password and	‘Password Changed’ ‘Forms’ in profile_test.dart.

	confirm new password, and press change password. Success message is displayed on successful update. This feature is tested by ‘Password Changed’ and ‘Forms’ in profile_test.dart.	
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.	There are multiple error messages with respective text describing the nature of the issue with users input. Each error message has its own test, some tests have various cases of invalid input that triggers the same error message. Error message tests are ‘Password requirements’, ‘Passwords don't match’, ‘Incorrect password’ and ‘New password same as old password’ in profile_test.dart.	-‘Password requirements’, ‘Passwords don't match’, ‘Incorrect password’ ‘New password same as old password’ in profile_test.dart.
Update details	**-**	**-**
Users must be able to change their username and school year.	This feature has been implemented. To change their details, users have to open a form, enter 1 or more of: school year, username, and press update details. Success message is displayed on successful update and ui updates with new details. This feature is tested by ‘details updated’, ‘1 details updated’ and ‘Forms’ in profile_test.dart.	-‘details updated’ in profile_test.dart. -‘1 details updated’ in profile_test.dart. -‘Forms’ in profile_test.dart.
Error messages must be present when the user enters invalid input. These error messages must describe the issue with the user's input.	There are multiple error messages with respective text describing the nature of the issue with users input. Each error message has its own test, some tests have various cases of invalid input that triggers the same error message. Error message tests are ‘No data’, ‘School year’, ‘School year decimal’, ‘Username length’ and ‘Username taken’ in profile_test.dart.	- ‘No data’, ‘School year’, ‘School year decimal’, ‘Username length’ ‘Username taken’ in profile_test.dart.

Introductory page	-	-
Introductory page must be present with a welcome message.	An introductory page has been implemented so there is a big container with a message and if users click it they are directed to login. This is tested by ‘container’ on intro_test.dart.	-‘container’ on intro_test.dart

figure 8.1. Functional requirements analysis

-Other tests have been performed that aren't directly to fulfil functional requirements. An example is ui components such as verifying app bars, page titles, containers etc. These tests don’t help reach functional requirements but they do help to verify the functionality of the app.

8.2. Original non-functional requirements and where they were met

Original requirements	How they were met	Test reference
Usability	-	-
The ui should be highly usable across all pages and features.	App is highly usable across all features apart from login and the forms on profile page, which are mostly usable but not highly usable. Before deployment, the app will be highly usable across all features.	6.6. Usability study
App should score highly in usability tests.	App did well in my usability study with the few problems noted and those that were due to fault in my application noted as changes that need to be made before deployment.	6.6. Usability study
Security	-	-
Password requirements must be met to create an account such as passwords must contain a sign and number.	These are some of the requirements for passwords. They are tested by ‘password’ on register_test.dart and Password requirements‘ on profile_test.dart.	-‘password’ on register_test.dart -‘Password requirements’ on profile_test.dart.

Users must be able to read/write only their own data.	My application allows users to read their data from profile page and question page. My application allows users to write their data from profile page, question page and registration page. My application doesn't allow users to read or write each other's data.	-
Database containing user data must be encrypted.	Firebase automatically encrypts data both in transit and at rest.	-
Passwords must be hashed.	Firebase automatically hashes passwords.	-
Other security measures may be added.	Password restrictions require passwords to have an upper case letter, a lowercase letter and to be at least 8 characters long.	-'password' on register_test.dart -'Password requirements' on profile_test.dart.
Performance	-	-
Features should be highly responsive with little to no wait time.	All features that require things to load are tested and timed in performace_test.dart. There are respective limits on how long these processes can take. These tests ensure that with good internet speed and a capable device there are no long wait times.	All of performace_test.dart
App should be able to hold 100 users as this is the capacity of the free plan on firebase.	Load testing is yet to be performed, however the app was designed and implemented to be able to support this user load and does indeed use Firebase's free plan.	-
Question generation should take less than a second per question with stable internet connection.	This is true and tested by 'Question Generation' in performace_test.dart.	-'Question Generation' in performace_test.dart
Age appropriate difficulty	-	-

Question difficulty for each school year should be researched and accurate to the uk national curriculum.	The question types/topics and ranges for numbers in said question types are specific to each learnage and based on the national curriculum for that school year.	5.4. Research / Appendix E: Text directly from national curriculum with chosen values
Scalability	-	-
App should be able to cope with an increasing amount of active users.	Firebase offers different packages that can support heavier user loads. If my current plan can't support the user load at a point in the future after deployment I would upgrade my plan.	-
App should be able to have updates where features or more questions are added.	I want to implement as much of my app as possible before deployment. This is so there are less big changes after deployment that could confuse my young target audience. Changes after deployment would likely be limited to adding and adjusting questions and question types.This will increase variety without adding options/functionalities that could confuse users. This means the app needs to be in a complete state before it can be deployed.	-
Reliability	-	-
Features should function as intended everytime they are used.	Various tests have been done on functional features to ensure functionality and reliability	6.2. Coded functional test plan with results
There should be no long loading screens	Every page is loaded and times recorded in 'Load pages', this is done alongside other functional features that require things to load and their combined results are outputted in 'Summary'. These tests are in performance_test.dart. If a page takes longer than 2 seconds to load, 'Load pages'	'Summary' in performance_test.dart / all of performance_test.dart

	will fail (other performance tests have respective time limits). If any performance test fails ‘Summary’ fails. Most pages usually load in less than a second but this is dependent on internet speed and device.	
Availability	_	_
App must ensure a minimum 99% uptime to ensure users have access to it.	This requirement isn’t measurable until after deployment. However the app would be fully developed before deployment and keeping a high uptime would be a primary goal.	-
Fault tolerance	_	_
App should be highly tolerant to user error without the ability of causing lasting damage to your account.	App is tolerant to user errors. I have analyzed this as part of my usability study in more detail. There are changes to be made before deployment that will further increase fault tolerance. Currently the only user errors that can cause lasting damage to accounts are: If the user forgets their password which will cause them to lose access to their account. I plan to resolve this before deployment. Depending on how you look at things, days in a row resetting could be considered lasting damage, however this is a core mechanic that I won't be considering as an issue.	6.6.3. Fault tolerance
Compliance	_	_
App must comply with all relevant data protection laws such as DPA 2018. App must comply with all university rules and regulations.	So far my project hasn’t involved real users or subjects. When I conduct a usability study on real users this will require research to ensure I comply with all relevant laws and regulations.	-

	Before deployment I will need to research laws and regulations regarding collecting and holding data to ensure I am doing so lawfully.	
Compatibility/accessibility	_	_
App must be compatible across different browsers and operating systems on pc.	App has been tested on firefox, edge and chrome on windows. It hasn't been tested on safari on ios but this is required before deployment.	-'Chrome' on compatibility_test.dart -'Edge' on compatibility_test.dart -'Firefox' on compatibility_test.dart
Quality assurance	_	_
Multiple testing procedures will be performed on the system to ensure everything is working as expected. This could include: usability testing, functionality testing, security testing, compatibility testing and difficulty testing.	Multiple test procedures have been performed including functionality testing, performance testing, compatibility testing, usability testing and a manual correctness and grammar check.	Chapter 6: Testing

figure 8.2. Non-functional requirements analysis

8.3. What's left

Due to limitations of the project (mostly time) there are things I haven't been able to add to my app/do yet but would have to before deployment. These are:

Features to add:

Multiple login error messages- I want to have 2 error messages for login, 1 for if email isn't linked to an account, and the other for if the password is incorrect. This will offer users better insight as to why they can't login.

Forgot password- I want to implement a forgot password feature that emails the user with a link to change their password. This improves fault tolerance as if users get locked out of their account they would be able to regain access, considering they have access to the email linked to the account.

Parental consent/verify account- Due to my young target audience I need to implement parental consent through a verification email on account registration. This also reduces the likelihood of bot accounts being created.

Delete account- I want to add a delete account feature to my app as it is important to give users this option.

Learnage on question page- Adding a display of the learnage (school year + 1 if in advanced mode) to the question page would increase the likelihood of users realising if/when they are doing questions at the wrong difficulty.

Issues to fix:

Takes a couple questions for learnage to update- If the school year or advanced mode is changed it takes 2 questions for the difficulty level to update. This is a functionality issue that requires fixing.

Profile page forms closing- The forms, update details and change password, close if the user enters invalid input. Combined with the fact that the app can only tell the user one issue everytime they fill in either of the forms even if more issues are present, could lead to a frustrating experience. To resolve this the forms should stay open until the user closes it or successfully fills it in. the users invalid input should also remain present so they don't have to retype things.

Change api key- due to file size I had to submit my work as a public git repository. This is insecure and needs to be changed before deply

Testing to complete:

Useability testing with real people- It is important to verify various aspects of usability using real people in my target audience. However due to my young target audience research would also need to be performed on performing an ethical study on children. I expect this testing will include general usability testing and difficulty testing.

Security testing- It is important to perform testing to ensure the security and safety of user data.

Ios testing on safari- I need to make sure the app runs on safari through ios to ensure accessibility to all major casual pc operating systems.

Load testing- This is necessary before deployment as it ensures the application will be able to support its users.

Updated functionality testing- If i make further implementation changes functionality testing would need to be updated in accordance.
Laws and regulations- Before deployment I need to ensure my application collects and holds users data lawfully.

After deploy:
Develop mobile version- This would improve accessibility. Both mobile and pc versions would need to be compatible with each other
Upgrade to paid firebase plan- If I am reaching the database capacity of the free firebase plan, I will need to upgrade to a paid plan.

Chapter 9: Conclusion

9.1. Aims and objectives

I completed all of my development objectives as part of my project. I have designed, implemented and tested my app so far, that when it's complete, my app will fulfill my objectives and aim. Due to the scale of my application It hasn't been fully developed yet however the progress achieved represents significant advancement toward these objectives and aim. It is hard to assess these in more detail while my development journey is still unfinished.

A personal goal of this project for me was to develop an application that comes as close to industry standards as I am currently capable of developing. Because of this I have been harsh on my app in a sense where I am trying to perfect everything. I am aware it is very difficult to develop an application at industry standard in a short amount of time, so I made the decision that I would rather have a proof of concept that is on its way to becoming an industry level app, than a 'completed app' that isn't really complete. I have avoided cutting corners and/or rushed solutions and I am very happy with all aspects of my project.

The literature review identified key elements of successful math apps that Math Club successfully implements: tracking progress (through score/10 and days in a row), instant visual feedback on answer, age appropriate difficulty and more. In my opinionI I successfully gained valuable information from performing this review and I managed to translate this into the following chapters of my project in order to better develop my app.

In terms of my requirements, I achieved the majority of them and still plan to achieve those that I haven't yet been able to. A lot of the changes in (7.3. What's left) are not to do with requirements they are to do with features being added after the requirements were written. The end of this project doesn't mark the end of development for my app so I will still be able to pursue requirements further. For how I achieved my functional and non-functional requirements in more detail see (7.1. Original functional requirements and where they were met) and (7.2. Original non-functional requirements and where they were met).

9.2. Risks

My project addressed risks identified in my initial log. Some of these risks lead to problems that had to be resolved:
laptop failure- I accidentally smashed my computer screen during development however it was repaired within 3 days of being smashed so this didn't really delay development.
Project too big- I predicted that the project may be too big to finish within the timeline and I was correct. I also increased the scope of the project. I made the decision to prioritise quality and I am ok with having not finished development.

Other risks were mitigated by careful planning and trying my best to stay ahead while progressing with all aspects of the project:
Not being able to program backend elements, Inadequate research, Missing deadlines, App not being functional, app not being usable, No time to test.

9.3. Development Experience

This is by far the largest and most complex project I have completed by myself. Developing and documenting my development journey was more difficult than I had anticipated.

I conducted a literacy review to learn how children in primary school interact with math apps, whether these apps are helpful, and if they improve grades. The review concluded that 90% of these apps contributed positively to improving children's mathematical learning and development. I also evaluated the strengths and weaknesses of existing apps to understand their effectiveness and areas I thought I could improve. The research also identified that personalised learning apps significantly improve learning outcomes. I learnt that effective apps prioritised usability, engagement, and alignment with learning objectives to support primary school children in developing math skills. The information I gained from my literacy review was used to aid the design of my app and form my requirements.

Design went as I had expected. I feel confident at interface design and I knew how I wanted my application to work. I didn't think I would find this chapter particularly challenging and I was correct in that evaluation.

Knowledge gained from the literacy review and my understanding of my design, allowed me to form requirements that define my application.

I would actually say implementation was easier than I had expected. I encountered problems during implementation but they were mostly resolved quicker than I had expected. I didn't particularly keep track of self set deadlines, however I do know that after I finished the implementation chapter I was ahead of where I wanted to be. This chapter was more difficult than design and I did find it highly challenging, but it was not as difficult as I had expected.

Testing started off well, I chose to do the easier tests first so I made a lot of progress quickly. After I had completed the easier tests, my progress slowed as tests were taking progressively longer to complete. Towards the end of functionality testing, some tests were taking over a day to complete. Some tests required me to change pages to make them test friendly or write code to support them, this was time consuming and not something I had planned to do. Some tests uncovered issues in my app that needed to be resolved so this led to more implementing. These factors led to testing taking longer and being more complicated than I had anticipated. I have 92 tests and ensuring correlation between my document and code was time consuming. I had to check my test tables to ensure every documented test accurately described its coded counterpart. Non coded tests such as my usability study and my grammar and correctness check weren't particularly difficult or time consuming to produce, however due to the constraint of time I wasn't able to implement the changes that came from my usability study into my app. As a whole this chapter was very time consuming and the coded tests were challenging to produce.

Documenting my development journey came with its challenges. A lot of the documenting wasn't done as I went along, and when I came back to do it I struggled to remember certain details. This mostly refers to the implementation chapter as I implemented everything before I documented any of it. Another challenge I faced was ensuring document consistency after changes. If I make a change to any aspect of my project, this requires me to cross check and make documentation changes so everything correlates. Some changes require more secondary changes than others, ensuring all these changes were made with no inconsistencies was time consuming and tedious. I also suffer with severe ADHD and dyslexia, because of this I struggle with reading, writing and concentrating. Despite this I feel like I have done well at documenting my development journey, and I am happy with my report.

Overall I feel good about how my development journey went.

9.4. Increasing Complexity

An issue I faced while developing my app was rising personal standards and addition of ideas I hadn't initially thought of or planned. While implementing and testing I came up with a lot of features to add and things to change. These ideas would help to improve my app. The issue is I didn't stop getting ideas and I found myself still implementing late in the testing chapter. Completing my usability study uncovered even more ideas and changes I had to make before I could deploy my app. In the end there wasn't enough time to implement all my ideas within the project deadline. Anything I do change would most likely require testing and multiple changes to the right up and there are multiple changes I would like to make. My solution to this was implementing the features that are most important first and getting as much done as I can before my project is due. Anything that could not be completed in time would be noted to be completed before deployment.

9.5. Project Management

I didn't necessarily follow my tasks and timescale from my PID however I still completed the project in a structured manner. First, I performed a literature review on the current state of maths and maths apps in primary education, including research. Next, I designed my app and then implemented it. The ranges for different ages in my question algorithm required researched values based on the national curriculum. After implementation, I tested my app. I mostly stayed on track, however during testing I was left with a heavier workload. I did parts of my writeup as I went along, however it wasn't finished until after testing. At the end of the project I was finalising my write up and code through small changes. (Appendix A: project initiation document, 10. Project tasks and timescales)

9.6. Product Assessment

I am happy with everything that is included in the current version of my app, all features are functional and usable. I am aware it is not complete however I know what needs to be done to complete it.

The current state represents a solid foundation with clear educational value. The app successfully achieves its core purpose of providing age-appropriate math practice through a "little and often" approach. The questions align well with the UK national curriculum, with question difficulties tailored to each year group. Comprehensive testing has validated the app's functionality, usability and more giving me confidence in the implementation (there are 92 coded tests as well as non coded tests).

Most importantly, the application stays true to my original vision of making math practice engaging and accessible for primary school children.

Reference list

DfE. (2025, February 27). Key stage 2 attainment. Gov.uk. https://explore-education-statistics.service.gov.uk/find-statistics/key-stage-2-attainment/2023-24

DfE. (2023, July 11). Attainment in maths and writing SATs increases compared to 2022. Gov.uk. https://www.gov.uk/government/news/attainment-in-maths-and-writing-sats-increases-compared-to-2022

Maths Apps: The Pros and Cons of Gamified Learning. (2023, February 7). Dr Gummer Good Play Guide. https://www.goodplayguide.com/blog/maths-apps-the-pros-and-cons-of-gamified-learning/

DfE. (2021, May). Education Technology (EdTech) Survey 2020-21 Research Report. Gov.uk. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1057817/Education_Technology_EdTech_Survey.pdf

Segundo, E. (2023, November 15). 3 REASONS WHY SOME KIDS DON'T LOVE MATH. Mathnasium The Math Learning Centre. https://www.mathnasium.com/math-centers/elsegundo/news/3-reasons-why-some-kids-dont-love-math

Nielsen, J. (1994, April 24). 10 Usability Heuristics for User Interface Design. nngroup https://www.nngroup.com/articles/ten-usability-heuristics/

Wong, E. (2025, March). Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces. Interaction design foundation. https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces?srsltid=AfmBOor1A5CERsxK2LMSrFwzI44XrJKmE1K5GRCcMC0MceMQkmwLaLJH

Bushell, M., & Greenway, J., & Palin, A., & Purvis, C., & Simpson, C. Free Primary Maths Online 10-Minute Tests - Years 1 to 5. CDG Books. https://www.cgpbooks.co.uk/resources/cgp-s-free-online-10-minute-tests/free-primary-maths-online-10-minute-tests-years-1?srsltid=AfmBOoojDLYMT9_4bmLKv-ViyM_OKsjzT1iEdpmUtmVSk_BmzMmJmP9o

Practice Math Games. Math Games. https://eng.mathgames.com/math-games.html

DfE. (2013, September). Mathematics programmes of study: key stages 1 and 2 National curriculum in England. gov.uk. https://assets.publishing.service.gov.uk/media/5a7da548ed915d2ac884cb07/PRIMARY_national_curriculum_-_Mathematics_220714.pdf

Privacy and Security in Firebase. (2025, April 9). Firebase. https://firebase.google.com/support/privacy

Firebase Authentication Password Hashing. (2025, April 9). Firebase. https://firebaseopensource.com/projects/firebase/scrypt

figure 0.1. Reference list

Appendices

Appendix A

Project Initiation Document

Project Details:

Project Title:	*Daily Maths app (may be subject to change)*
Student:	*Harry Miller*
Course:	*Computing*
Project code:	*PJE40 for an engineering project*
Supervisor:	*Amanda Peart*
Date:	*14/10*

Declarations (Please tick)

y h ☑	I give permission for this document to be made available to other students as examples of previous work. (optional).

yh ☑	I confirm that I have read and understood the University Rules in respect of plagiarism and student misconduct.
yh ☑	I declare that this work is entirely my own. Each quotation or contribution cited from other work is fully referenced.

1. Target Audience

My app will be aimed at primary school children. This means the app has to be child friendly in every sense. This app could be used as a curricular or extracurricular activity. It will be a web application, to ensure that children can access it from school computers, and won't be excluded due to any living situation or lack of a phone. You could also say that schools are a target.

Aspects of design will aim to appeal to the target audience's personal preferences, such as: colour scheme, images present.

Other aspects of design will aim to maximise usability for the target audience, such as: large font sizes, clear text that all ages can understand, minimal reading, simple vocabulary, correct question difficulty for all ages, simple interface and navigation, limited options to not overwhelm users, colour prompt for correct/incorrect answers.

Some text pages are aimed at parents and teachers as opposed to students (about us and FaQ). The purpose of these is to educate users on the application their students/children are using. These pages will be more text based and aimed at adults.

2. Degree suitability

My project is relevant to my computing degree because I am choosing to make an app using skills I have learnt on my course. These may include but are not limited to: programming, usability analysis, design and implementation, security, testing, software engineering.

I can look back at my modules and see how some of them are very relevant to small areas of my project, but when I bring them all together they have taught me the skill set I need to design and produce an app/apps.

The idea for my app isn’t related to my degree but the fact I'm making it and the skills I plan to use are.

3. The project environment and problem to be solved

I am trying to solve the issue of primary level kids not enjoying or doing their school work (education is a big topic so I will focus on maths only). I believe that one of the biggest distractions for children is technology. I want to find a way to distract early development children with educational technology, so they can build positive habits and fundamentals from a younger age. Although this app is aimed at primary school kids, the aim would be to set children up for a better future and hopefully build better working habits throughout their education.

There are already maths apps children can use however I want to make mine differently. Another reason why I chose to make this app for primary school ages is because the majority of the educational apps are aimed at secondary school children and aren't user friendly to early years. They have too many options which may confuse younger children. In my app the user will only need to worry about the current question and another question will appear after.

Another difference between my project and other maths apps is the length of time it takes to complete tasks. My app will keep track of how many days in a row you have completed and to achieve completion for any given day the user only has to answer 10-22 questions depending on their age and mode. This should take most users between 5-10 minutes as questions will be generated at different difficulties appropriate for each age group. Other apps' daily tasks are too long, or apps don't have a daily completion feature so users aren't given a point to stop. Either could make the app feel like it takes too long or causes hassle, and disincentive users from returning.

I believe these differences should better direct the model of a maths app, to the target audience of primary school users.

4. Project aim and objectives

Aim: The aim for this project is to produce a maths application that incentivises children to do more maths little and often. The app should help children build a habit out of doing maths

Objectives:

- Help children who would prefer to use computers than paper
- Help children enjoy maths more
- Help children learn and revise maths fundamentals
- Help children achieve higher grades in maths
- Help children put more overall time into there maths studies
- Give teachers and parents more productive computer tasks for children

Development Objectives;

- Completing pid and ethics form and getting them approved
- Literature review primary school maths

- Evaluate maths applications
- Determine requirements
- Design application
- Implement application based off design
- Test functionality and usability of application
- Evaluate project and write report
- Prepare for presentation

5. Project constraints

Some constraints I have are: limited support, limited knowledge, team of 1. This is a difficult project so these constraints suggest that I have a lot of work to do which could even incorporate having to learn new skills. Taking this into consideration I will work to get ahead so if/when I do encounter issues, I have time to resolve them even if it takes longer than i would want.

Another constraint is deadlines.This project has multiple deadlines:
Choosing your Project Topic and Supervisor- deadline 11th October 2024
Project Initiation - deadline 25th October 2024
Ethical Approval - deadline 6th December 2024
Satisfactory Progress Presentation - deadline 31st January 2025
Final project submission - deadline may 2025
Final Project Presentation - deadline June 2025
Also other modules have deadlines I need to be ready for. I can get around these deadlines by planning my time and balancing.

The devices I will use on my project are my laptop and the uni PC's. These are not subject to change however i wouldn't call them constraints as they can do everything I need them to do.

The resources on moodle are constrained by quantity so if I need something that Isn't there I will need to search the internet. At this point my digital resources are constrained by quality.

6. Facilities and resources

- Laptop. No constraints as long as my laptop is functional.
- Uni PC. Constraint on me needing to be somewhere that has uni PC's.
- Library. Constraint on how much time I will be able to spend there.
- My knowledge. Limited to what I know and have learned(mostly on this course).
- Module and course resources on moodle
- Relevant development software
- Any information I find on the internet. Constraint as not all information is usable and reliable.
- chatgpt / co pilot

- Amanda Peart and maybe other staff. Constraint on how much support I will be able to receive.

7. Log of risks

No	Description	Likelihood (high, medium, low)	Impact	Mitigation/Avoidance
1	Laptop failure	Medium	Cause delays to the project timeline and data loss	Back up work to a USB and another source
2	Breaking laws regarding data, childrens data and security	Low	Possible legal situation	Research what i can and can't do
4	Missing deadlines	High	Capped or fail module	Stay on top of my work and plan ahead and with extra time
5	App not being usable regarding functionality	High	Low grade	Make sure i have enough time to program all features
6	App not being usable regarding GUI	Low	Low	Put time into making the app look nice
7	Not being able to program backend elements	High	Low grade	Make sure i have enough time to program all features
8	No time to test	High	Low grade	Leave time to do detailed tests
9	Inadequate research	High	-Low grade -Break laws -unusable app	Make sure I get enough reliable research
10	App not appealing to target audience	High	Waste of time	Try my best to make the app appeal to the target audience
11	Lack of feedback	Medium	App developed from one perspective	Get as much feedback as i can from my project supervisor
12	Project too big	Medium	-unfinished app -low grade	Leave enough time to make it

13	Myself	medium	-unfinished app - low grade	- Use slippage time

8. Project deliverables

Plan/spec documents including
- Designs
- Research references

Ethics certificate
Project code
presentation
Literature review primary school maths
Evaluate maths applications

9. Project approach

My project needs to be well planned and organised and to do this I will need a structured approach. I need to manage my time effectively and prioritise tasks

to meet deadlines when necessary. I will do this by breaking the project into small achievable milestones. I will need to try and identify potential obstacles and where possible, prepare solutions in advance. I am aware that to achieve these targets I will need to have regular check-ins with my project supervisor.

I will make my app on visual studio code using Flutter/Dart as my primary language. I will implement firebase to support the backend.

During my literature review I plan on researching the current state of math and math apps in relation to primary education including studying other maths applications. This will help me better understand the topic and what makes a good maths app.
When I Implement my question generation algorithm I will need to research the values it will use for questions at different ages. This will ensure that question difficulty is accurate to respective school years.

10. Project tasks and timescales

No	Stage	Dates	Main Tasks
1	Project start up	18/09/23 - 02/10/23	-Choose topic and supervisor
2	Project plan	02/10/24- 02/10/31	-Project initiation document, -Design, -App architecture plan
3	Implementation -navigation and interfaces that don't require backend	02/10/24- 02/10/31	-Navigation, -About us, -Faq
4	ethics review	By 6/12/2024	-Research ethics, -Produce any required documents,

			-apply for ethics review
5	Continue implementing frontend and backend	By 22/1/2025	-Questions page ui and algorithms with placeholder values, -Profile page, -Login/create account/log out, -Storing user data -fill in values for about us and faq
6	Research laws and primary school maths and apply research to app through questions.	By 31/2/2025	-Research questions for every age group, -Research laws that may be an issue -Apply research to question page code by updating values, -Tidy up code and fix issues
7	Apply research to app through questions and complete implementation	By 31/2/2025	-Apply research to question page code by updating values,
8	Tidy code	By 15/3/2025	-Tidy up code and fix issues
9	testing	By 31/4/2025	-Make test plan based on app, -Complete tests from test plan
10	presentation	By 31/5/2025	-Plan presentation, -Perform presentation

11. Supervisor meetings

Amanda Peart / Weekly in person / 11am thursday / PO2.27

12. Legal, ethical, professional, social issues

Legal:

- Data protection: General Data Protection Regulation, which imposes strict regulations on collecting data from minors.
- Intellectual Property: ensuring that content does not infringe on copyrights or trademarks.
- Accessibility standards: compliance with laws requiring that educational materials be accessible to all students, including those with disabilities (Equality Act 2010).

Ethical:

- Data security and privacy: implementing measures to protect sensitive information such as encryption.
- Fairness and bias: ensuring that the content is inclusive and does not reinforce stereotypes or biases.

Professional:

- Responsibility to users: maintaining a commitment to providing accurate information and a safe learning environment for children.
- User centred design: prioritising the needs and experiences of primary school students in the design and development process.
- Follow bcs code of conduct

Social:

- Digital Divide: Addressing potential inequalities in access to technology and the internet that may affect the app's usability.

figure 0.2. Appendix A: project initiation document

Appendix B
Ethics certificate

Certificate of Ethics Review

Project title: maths app

Name:	harry miller	**User ID**:	up2120303	**Application date**:	07/11/2024 14:25:44	**ER Number**:	TETHIC-2024-109663

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou**, **Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **amanda peart** Is the study likely to involve human subjects (observation) or participants?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

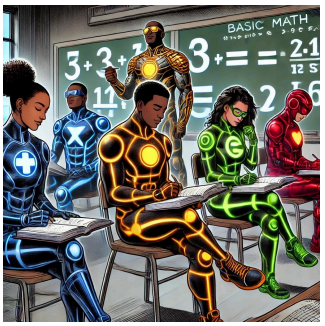
Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy. Supervisor comments:

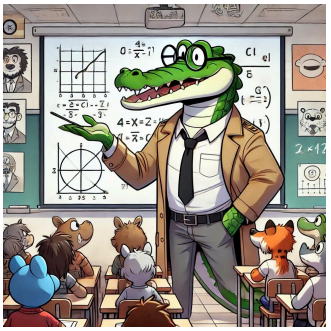
Supervisor's Digital Signature**: amanda.peart@port.ac.uk** Date**: 28/04/2025**

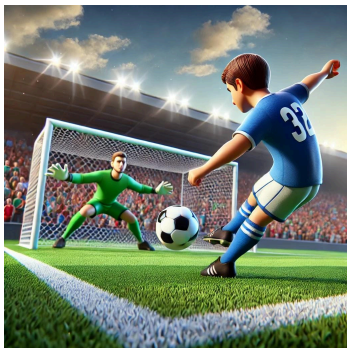
figure 0.3. Appendix B: Ethics certificate

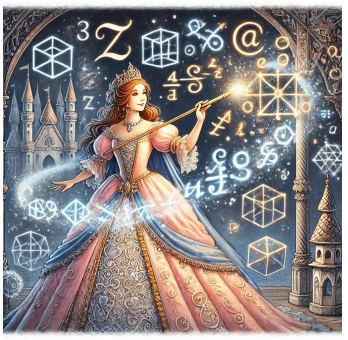
Appendix C
Images used in Math Club

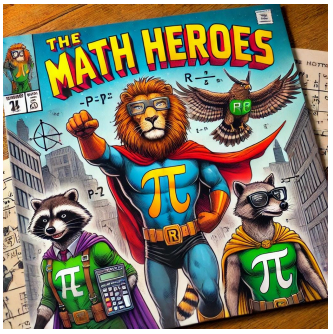





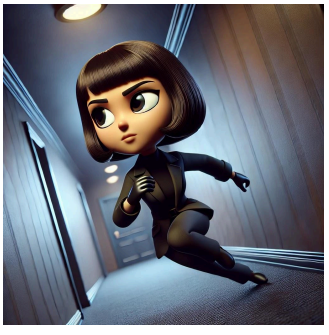










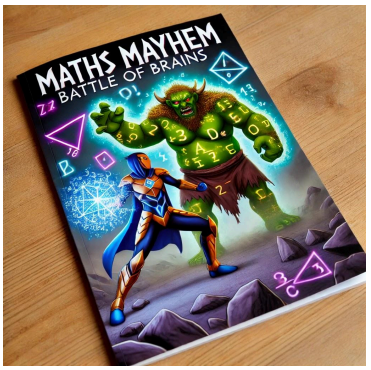


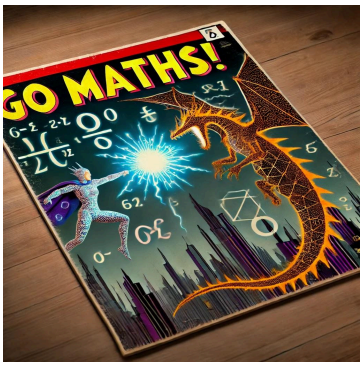

hath



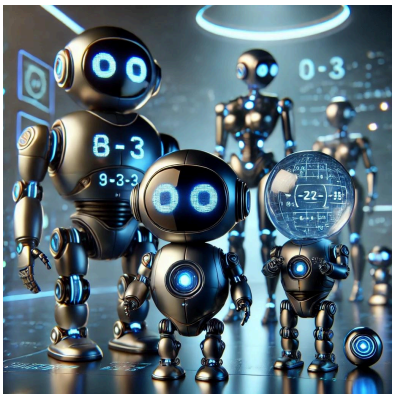




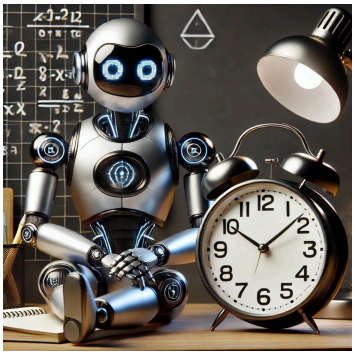


MATHS MAYHEM
BATTLE OF BRAINS


GO MATHS!


professor


8-3
0-3


3



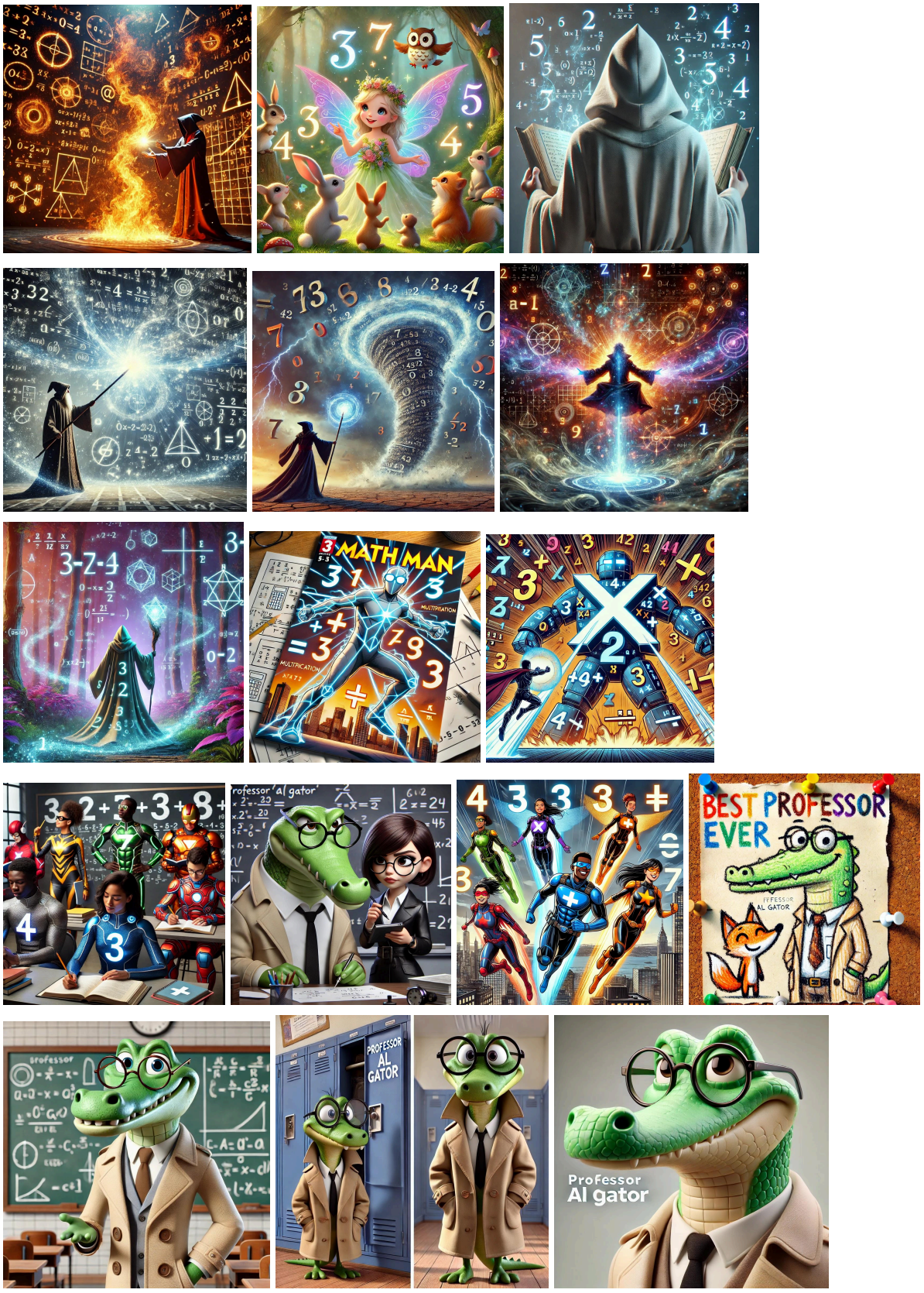


figure 0.4. Appendix C: Images used in Math Club

Appendix D
Worded questions

Questions have been changed, added and removed in code but this table hasn't been updated. After checking my code some of the questions had issues that will later be resolved in testing. I'm not sure when the issues occurred however they are not present in the most recent version of my app.

Year 1 (25/25)

Select the missing number: 1, 2, 3, _, 5	4
Select the missing number: 3, 4, _, 6, 7	5
How many sides does a square have?	4
How many sides does a triangle have?	3
Sam has 3 pens. Sam loses a pen. How many pens does Sam have left?	2
How many tens are in 20?	2
Which number has 2 tens and 3 ones?	23
What is the missing number: 9 = ? + 5	4
What is the total of 2, 3 and 4?	9
What is 6 more than 3?	9
How many sides does a rectangle have?	4
What number has 1 ten and 7 ones?	17
What is 5 more than 3?	8
Select the missing number 5, 6, 7, _, 9	8
Claire and Zoe have 4 sweets each. How many sweets do they have altogether?	8
What is the missing number: ? - 3 = 2	5
How many must be added to 9 to get 16?	7
Sue buys 10 bananas. She eats 2 of them. How many are left?	8
What is half of 12?	6

Which number is twenty-seven?	27
What number is 3 more than 5?	8
What number is 7 less than 9?	2
What number is 7 more than 20?	27
Which number is added to 14 to make 20?	6
What is the next number in this sequence? 2, 4, 6, 8, __	10

Year 2 (25/25)

How many sides does a pentagon have?	5
How many sides does a hexagon have?	6
How many days in 1 week?	7
Which number is missing: 10 + ? = 100	90
Which number is missing: 17 + ? = 20	3
What is 10 groups of 3?	30
How many 10's are there in 50?	5
Which number is missing? 20 ÷ ? = 4	5
Which number is missing? ? ÷ 2 = 7	14
5 children have 5 sweets each. How many sweets do they have in total?	25
What is 5 groups of 4?	20
How many seconds in one minute?	60
How many hours in a day?	24
How many 5's are there in 30?	6
How many minutes are there in a quarter of an hour?	15
What is the total of 2, 15 and 3?	20

What is the total of 7 and 26?	33
What is 5 + 5 + 10 + 5 + 5?	30
What is the sum of 7, 9 and 3?	19
A table has 4 legs. How many legs do 4 tables have?	16
4 friends share 8 slices of pizza. How many slices will they each get?	2
What is the missing number: 20 - ? = 8	12
If one child has 10 fingers, how many fingers will 4 children have?	40
How many tens are there in 170?	17
What is the missing number: 14 - ? = 2	12

Year 3 (25/25)

What number is missing? 8 x ? =88	11
How many sides does an octagon have?	8
How many sides does a hexagon have?	6
How many sides does a Pentagon have?	5
An apple weighs 25g. How many grams would 4 apples weigh?	100
6 bricks weigh 600g. How many grams do 2 bricks weigh?	200
How many pence is £7.96?	796
How many days in 3 weeks?	21
What is 400 – 100 – 132?	168
How many seconds in 3 minutes?	180
What is the result of 40 divided by 10?	4
What is 9 times 5?	45

How many faces does a cube have?	6
How many 6s in 36?	6
What number do I get if I share 45 between 5?	9
What is the next number in this sequence? 7, 14, 21, 28	35
What is the next number in this sequence? 18, 24, 30, 36	42
What number is missing: 12 x ? = 36	3
What is the next number in this sequence? 8, 16, 24, 32	40
What number lies halfway between 100 and 200?	150
What number is missing: 42 - ? = 29	13
What is the next number in this sequence? 4, 8, 13, 19	26
What number lies halfway between 40 and 70?	55
What number lies halfway between 20 and 100?	60
How many 5's in 60?	12

Year 4 (25/25)

What would we add to 56 to get 100?	46
Which number comes next? 22, 27, 32, 37	42
Which number lies halfway between 1,020 and 1,050?	1035
What is one tenth of 30?	3
What is one quarter of 60?	15
Which multiple of 4 comes after 32?	36

Which number is divisible by 4 and 5?	20
What is the next number of the sequence? 136, 140, 144, 148	152
How many degrees is one whole turn?	360
How many degrees in half of one whole turn?	180
What is one third of 150?	50
What is one quarter of 100?	25
What is one quarter of 500?	125
What is one third of 21?	7
What is one third of 63?	21
What is half of 340?	170
What number is missing: 15 x ? = 90	6
What number is missing: ? – 32 = 32	64
What number is divisible by 3 and 6?	18
What is one fifth of 100?	20
What is one third of 36?	12
What is one tenth of 130?	13
How many degrees in a right angle?	90
What's the product of 6 and 9?	54
A box has 5 rows of 8 apples. How many apples are there in the box?	40

Year 5 (25/25)

Which number is halfway between 27,400 and 28,000?	27700
What is the next number of the following sequence? 64, 56, 48, 40, 32	24
What will be the next number of this sequence? 22, 33, 44, 55, 66	77
What needs to be added to 45,600 to make 45,912?	312
What is 2 fifths of 100?	40
What needs to be subtracted from 4,733 to make 2,723?	2010
What is the next number of the sequence? 9, 18, 27, 36	45
What will be the tenth number of this sequence? 11, 22, 33, 44, 55, 66	121
What's the product of 7 and 8?	56
What is the next number in this sequence? 14, 28, 42, 56	70
What is three quarters of 72?	54
What is two tenths of 55?	11
What is the next number in this sequence? 60, 120, 240, 480	960
What is the next number in this sequence? 150, 300, 450, 600	750
What is 4 tenths of 200?	80
In every 10 bricks, 3 are red. We have 30 bricks so how many are red?	9
A box holds 12 eggs. I have 168 eggs. How many boxes do I have?	14
What is one third of 90?	30
What is a fifth of 120?	24

What number is missing: 152 - ? = 135	17
What number is missing: 81 x ? = 243	3
What is 3 fifths of 250?	24
What is 2 sixths of 240?	80
A can holds 50 beans. How many cans are needed for 1000 beans?	20
If I halve 64 three times, what number do I have left?	8

Year 6 (25/25)

How many faces does a square-based pyramid have?	5
What is 450 rounded to the nearest 100?	500
Which is the mode of this data: 2, 5, 7, 2 , 7, 2, 3?	2
What is the mean of this data: 11, 14, 17, 18, 20?	16
What is the median of this data: 11, 14, 17, 18, 20?	17
What is the range of this data: 11, 14, 17, 18, 20?	9
What is the next number of the sequence? 1, 3, 6, 10, 15, 21	28
How many sides on a Decagon?	10
What is multiplied by 4 to give 180?	45
Which is the median in this set of data: 2, 4, 4, 6, 8, 10, 16?	6
Which is the mode in this set of data: 2, 4, 4, 6, 8, 10, 16?	4
What is the mode of this data: 2, 3, 3, 5, 5, 5, 7	5

What is the mean of this data: 2, 3, 3, 5, 5, 6	4
What is the median of this data: 2, 3, 3, 5, 5, 5, 7	5
What is divided by 6 to leave 9?	54
What is one sixth of 132?	22
What is 2 sevenths of 140?	40
What is three quarters of 160?	120
What number is missing: 131 + ? = 277	146
What number is missing: 840 - ? = 85	755
What is 3 eighths of 320?	120
What is 2 twelves of 72?	14
What is 4 fifths of 300?	240
How many faces does a dodecahedron have?	12
What is the next number of the sequence? 20, 10, 30, 20, 40	30

Year 6 Advanced (15/15)

Normal year 6 worded questions as well as:

What is the mean of this data: 32, -8, 14, 40, 22?	20
What is the median of this data: 32, -8, 14, 40, 22?	22
If a tap drips once every 15 seconds, how many times will it drip in an hour?	240
What is the mean 22, 23, 23, 25, 27, 30	25
What is the mode 22, 23, 23, 25, 27, 30	23
What is the median 22, 23, 23, 25, 27, 30	24

How many hours is one week?	168
How many minutes in a day?	1440
What is the next number of the sequence? 23, 30, 38, 47, 57	68
What are three eighths of 888?	333
The 4th and 5th multiples of a number are 24 and 30. What is the number?	6
Which of the following is a common factor of 15, 30, 45 and 20?	5
The 2nd and 6th multiples of a number are 22 and 66. What is the number?	11
The 3rd and 8th multiples of a number are 42 and 112. What is the number?	14
What is the sixth term in the sequence with Tn = 2n - 5?	7

figure 0.5. Appendix D: Worded questions

Appendix E
Text directly from national curriculum with chosen values (DfE, 2013)

Year 1

Pupils should be taught to: • count to and across 100, forwards and backwards, beginning with 0 or 1, or from any given number • count, read and write numbers to 100 in numerals; count in multiples of twos, fives and tens • given a number, identify one more and one less • identify and represent numbers using objects and pictorial representations including the number line, and use the language of: equal to, more than, less than (fewer), most, least • read and write numbers from 1 to 20 in numerals and words. **Statutory requirements** Pupils should be taught to: • read, write and interpret mathematical statements involving addition (+), subtraction (–) and equals (=) signs • represent and use number bonds and related subtraction facts within 20 • add and subtract one-digit and two-digit numbers to 20, including zero • solve one-step problems that involve addition and subtraction, using concrete objects and pictorial representations, and missing number problems such as 7 =– 9. **Multiplication and Division NOT APPLICABLE** Statutory requirements Pupils should be taught to: • solve one-step problems involving multiplication and division, by calculating the answer using concrete objects, pictorial representations and arrays with the support of the teacher. **Notes and guidance (non-statutory)** Through grouping and sharing small quantities, pupils begin to understand: multiplication	**Adding and Subtraction: 10 and 10**

and division; doubling numbers and quantities; and finding simple fractions of objects, numbers and quantities. They make connections between arrays, number patterns, and counting in twos, fives and tens.	

Year 2

Statutory requirements Pupils should be taught to: • solve problems with addition and subtraction: • using concrete objects and pictorial representations, including those involving numbers, quantities and measures • applying their increasing knowledge of mental and written methods • recall and use addition and subtraction facts to 20 fluently, and derive and use related facts up to 100 • add and subtract numbers using concrete objects, pictorial representations, and mentally, including: • a two-digit number and ones • a two-digit number and tens • two two-digit numbers • adding three one-digit numbers • show that addition of two numbers can be done in any order (commutative) and subtraction of one number from another cannot • recognise and use the inverse relationship between addition and subtraction and use this to check calculations and solve missing number problems. **Notes and guidance (non-statutory)** Pupils extend their understanding of the language of addition and subtraction to include sum and difference.	**Adding and Subtraction: 20 and 100** **Multiplication and Division: 5 and 10**

Pupils practise addition and subtraction to 20 to become increasingly fluent in deriving facts such as using 3 + 7 = 10; 10 – 7 = 3 and 7 = 10 – 3 to calculate 30 + 70 = 100; 100 – 70 = 30 and 70 = 100 – 30. They check their calculations, including by adding to check subtraction and adding numbers in a different order to check addition (for example, 5 + 2 + 1 = 1 + 5 + 2 = 1 + 2 + 5). This establishes commutativity and associativity of addition. Recording addition and subtraction in columns supports place value and prepares for formal written methods with larger numbers. **Statutory requirements** Pupils should be taught to: • recall and use multiplication and division facts for the 2, 5 and 10 multiplication tables, including recognising odd and even numbers • calculate mathematical statements for multiplication and division within the multiplication tables and write them using the multiplication (×), division (÷) and equals (=) signs • show that multiplication of two numbers can be done in any order (commutative) and division of one number by another cannot • solve problems involving multiplication and division, using materials, arrays, repeated addition, mental methods, and multiplication and division facts, including problems in contexts. **Notes and guidance (non-statutory)** Pupils use a variety of language to describe multiplication and division. Pupils are introduced to the multiplication tables. They practise to become fluent in the	

2, 5 and 10 multiplication tables and connect them to each other. They connect the 10 multiplication table to place value, and the 5 multiplication table to the divisions on the clock face. They begin to use other multiplication tables and recall multiplication facts, including using related division facts to perform written and mental calculations. Pupils work with a range of materials and contexts in which multiplication and division relate to grouping and sharing discrete and continuous quantities, to arrays and to repeated addition. They begin to relate these to fractions and measures (for example, 40 ÷ 2 = 20, 20 is a half of 40). They use commutativity and inverse relations to develop multiplicative reasoning (for example, 4 × 5 = 20 and 20 ÷ 5 = 4).	

Year 3

Statutory requirements Pupils should be taught to: • add and subtract numbers mentally, including: • a three-digit number and ones • a three-digit number and tens • a three-digit number and hundreds • add and subtract numbers with up to three digits, using formal written methods of columnar addition and subtraction • estimate the answer to a calculation and use inverse operations to check answers • solve problems, including missing number problems, using number facts, place value, and more complex addition and subtraction. **Notes and guidance (non-statutory)**	**Addition and Subtraction: 100 and 100** **Multiplication and Division: 10 and 12**

Pupils practise solving varied addition and subtraction questions. For mental calculations with two-digit numbers, the answers could exceed 100. Pupils use their understanding of place value and partitioning, and practise using columnar addition and subtraction with increasingly large numbers up to three digits to become fluent (see Mathematics Appendix 1). **Statutory requirements** Pupils should be taught to: • recall and use multiplication and division facts for the 3, 4 and 8 multiplication tables • write and calculate mathematical statements for multiplication and division using the multiplication tables that they know, including for two-digit numbers times one-digit numbers, using mental and progressing to formal written methods • solve problems, including missing number problems, involving multiplication and division, including positive integer scaling problems and correspondence problems in which n objects are connected to m objects. Mathematics – key stages 1 and 2 **Notes and guidance (non-statutory)** Pupils continue to practise their mental recall of multiplication tables when they are calculating mathematical statements in order to improve fluency. Through doubling, they connect the 2, 4 and 8 multiplication tables. Pupils develop efficient mental methods, for example, using commutativity and associativity (for example, 4 × 12 × 5 = 4 × 5 × 12 = 20 × 12 = 240) and multiplication and division facts (for example, using 3 × 2 = 6, 6 ÷ 3 = 2 and 2 = 6 ÷ 3) to derive related	

facts (for example, 30 × 2 = 60, 60 ÷ 3 = 20 and 20 = 60 ÷ 3). Pupils develop reliable written methods for multiplication and division, starting with calculations of two-digit numbers by one-digit numbers and progressing to the formal written methods of short multiplication and division. Pupils solve simple problems in contexts, deciding which of the four operations to use and why. These include measuring and scaling contexts, (for example, four times as high, eight times as long etc.) and correspondence problems in which m objects are connected to n objects (for example, 3 hats and 4 coats, how many different outfits?; 12 sweets shared equally between 4 children; 4 cakes shared equally between 8 children).	

Year 4

Statutory requirements Pupils should be taught to: add and subtract numbers with up to 4 digits using the formal written methods of columnar addition and subtraction where appropriate estimate and use inverse operations to check answers to a calculation solve addition and subtraction two-step problems in contexts, deciding which operations and methods to use and why. **Notes and guidance (non-statutory)** Pupils continue to practise both mental methods and columnar addition and subtraction with increasingly large numbers to aid fluency (see Mathematics Appendix 1). **Statutory requirements**	**Addition and Subtraction: 1000 and 1000** **Multiplication and Division: 12 and 12**

Pupils should be taught to: • recall multiplication and division facts for multiplication tables up to 12 × 12 • use place value, known and derived facts to multiply and divide mentally, including: multiplying by 0 and 1; dividing by 1; multiplying together three numbers • recognise and use factor pairs and commutativity in mental calculations • multiply two-digit and three-digit numbers by a one-digit number using formal written layout • solve problems involving multiplying and adding, including using the distributive law to multiply two digit numbers by one digit, integer scaling problems and harder correspondence problems such as n objects are connected to m objects. **Notes and guidance (non-statutory)** Pupils continue to practise recalling and using multiplication tables and related division facts to aid fluency. Pupils practise mental methods and extend this to three-digit numbers to derive facts, (for example 600 ÷ 3 = 200 can be derived from 2 x 3 = 6). Mathematics – key stages 1 and 2 26 Pupils practise to become fluent in the formal written method of short multiplication and short division with exact answers (see Mathematics Appendix 1). Pupils write statements about the equality of expressions (for example, use the distributive law 39 × 7 = 30 × 7 + 9 × 7 and associative law (2 × 3) × 4 = 2 × (3 × 4)). They combine their knowledge of number facts and rules of arithmetic to solve mental and written calculations for example, 2 x 6 x 5 = 10 x 6 = 60. Pupils solve two-step problems in contexts, choosing the appropriate operation, working	

with increasingly harder numbers. This should include correspondence questions such as the numbers of choices of a meal on a menu, or three cakes shared equally between 10 children.	

Year 5

Statutory requirements Pupils should be taught to: • add and subtract whole numbers with more than 4 digits, including using formal written methods (columnar addition and subtraction) • add and subtract numbers mentally with increasingly large numbers • use rounding to check answers to calculations and determine, in the context of a problem, levels of accuracy • solve addition and subtraction multi-step problems in contexts, deciding which operations and methods to use and why. **Notes and guidance (non-statutory)** Pupils practise using the formal written methods of columnar addition and subtraction with increasingly large numbers to aid fluency (see Mathematics Appendix 1). They practise mental calculations with increasingly large numbers to aid fluency (for example, 12 462 – 2300 = 10 162). **Statutory requirements** Pupils should be taught to: • identify multiples and factors, including finding all factor pairs of a number, and common factors of two numbers • know and use the vocabulary of prime numbers, prime factors and composite (non	**Addition and Subtraction: 5000 and 5000** **Multiplication and Division: 12 and 100**

prime) numbers • establish whether a number up to 100 is prime and recall prime numbers up to 19 • multiply numbers up to 4 digits by a one- or two-digit number using a formal written method, including long multiplication for two-digit numbers • multiply and divide numbers mentally drawing upon known facts • divide numbers up to 4 digits by a one-digit number using the formal written method of short division and interpret remainders appropriately for the context • multiply and divide whole numbers and those involving decimals by 10, 100 and 1000 Mathematics – key stages 1 and 2 • recognise and use square numbers and cube numbers, and the notation for squared (2) and cubed (3) • solve problems involving multiplication and division including using their knowledge of factors and multiples, squares and cubes • solve problems involving addition, subtraction, multiplication and division and a combination of these, including understanding the meaning of the equals sign • solve problems involving multiplication and division, including scaling by simple fractions and problems involving simple rates. **Notes and guidance (non-statutory)** Pupils interpret non-integer answers to division by expressing results in different ways according to the context, including with remainders, as fractions, as decimals or by rounding (for example, 98 ÷ 4 = 4 98 = 24 r 2 = 24 2 1 Pupils use and explain the equals sign to indicate equivalence, including in missing	

number problems (for example, 13 + 24 = 12 + 25; 33 = 5 x). = 24.5 ≈ 25). Pupils use multiplication and division as inverses to support the introduction of ratio in year 6, for example, by multiplying and dividing by powers of 10 in scale drawings or by multiplying and dividing by powers of a 1000 in converting between units such as kilometres and metres. Distributivity can be expressed as a(b + c) = ab + ac. They understand the terms factor, multiple and prime, square and cube numbers and use them to construct equivalence statements (for example, 4 x 35 = 2 x 2 x 35; 3 x 270 = 3 x 3 x 9 x 10 = 92 x 10).	

Year 6

Statutory requirements	**Addition and Subtraction: 9999 and 9999** **Multiplication and Division: 99 and 99**
Pupils should be taught to: • multiply multi-digit numbers up to 4 digits by a two-digit whole number using the formal written method of long multiplication • divide numbers up to 4 digits by a two-digit whole number using the formal written method of long division, and interpret remainders as whole number remainders, fractions, or by rounding, as appropriate for the context • divide numbers up to 4 digits by a two-digit number using the formal written method of short division where appropriate, interpreting remainders according to the context • perform mental calculations, including with mixed operations and large numbers	**Advanced Multiplication and Division: 9999 and 99**

identify common factors, common multiples and prime numbers use their knowledge of the order of operations to carry out calculations involving the four operations solve addition and subtraction multi-step problems in contexts, deciding which operations and methods to use and why Mathematics – key stages 1 and 2 40 **Statutory requirements** solve problems involving addition, subtraction, multiplication and division use estimation to check answers to calculations and determine, in the context of a problem, an appropriate degree of accuracy. **Notes and guidance (non-statutory)** Pupils practise addition, subtraction, multiplication and division for larger numbers, using the formal written methods of columnar addition and subtraction, short and long multiplication, and short and long division (see Mathematics Appendix 1). They undertake mental calculations with increasingly large numbers and more complex calculations. Pupils continue to use all the multiplication tables to calculate mathematical statements in order to maintain their fluency. Pupils round answers to a specified degree of accuracy, for example, to the nearest 10, 20, 50 etc., but not to a specified number of significant figures. Pupils explore the order of operations using brackets; for example, 2 + 1 x 3 = 5 and (2 + 1) x 3 = 9. Common factors can be related to finding equivalent fractions.	

figure 0.6. Appendix E: Text directly from national curriculum with chosen values

Appendix F
Readme

```
Readme

Running application
open file in visual studio coad and run 'flutter pub get' then 'flutter
run' in the terminal.

Guide to use Math Club

Introduction: When you first open Math Club, click anywhere on the
welcome screen container to continue to the login page.

Login: Enter your email and password, then click "Login". If error
messages are present then reply to them and click "Register" again. If
you don't have an account yet, click "Don't have an account? Register"
at the bottom. Successful login directs users to home/question page.

Registration: If you dont have an account please enter:
Email (must be valid format, must be unique)
Username (4-12 characters, must be unique)
School Year (1-6, whole number)
Password (8+ characters with a number, uppercase, lowercase, and
symbol)
Confirm Password (must match)
Then click "Register", if error messages are present then reply to them
and click "Register" again. Successful registration directs users to
home/question page.

Awsering question: After logging in or registering, you'll see a math
question with three possible answers, click the answer you think is
correct.
You'll get instant feedback showing if you were right or wrong and the
correct awnser. To continue to the next question, click one of the
awnser boxes again or wait 3 seconds.

Score and days in a row: Your score (out of 10) shows at the side of
the home/question page. Awnsering questions correctly increments score
by 1 but score cannot increase past 10. Container containing score will
also change colour as score increases.
When score hits 10 days in a row increments by 1. Score resets every 24
hours at 12am. If score doesn't = 10 on reset or if the user doesn't
login for any given day, days in a row will reset with the score.
```

Highest days in a row is also recorded and updates with the value of days in a row if: days in a row > highest days in a row. Highest days in a row is never reset.

Navigation
From home/question page press 'menu' and then one of the dropdown options (Profile, About Us or FAQ) to navigate to respective page. Alternatively press Logout to logout and be directed to login. The Profile, About Us and FAQ page all have a back button which directs users back to homepage. The FAQ page has links to other pages where relevant to questions. Pressing back after using these links will direct users back to FAQ.

Profile Page
from the profile page you can view their profile details (email, username, days in a row, highest days in a row, advanced mode). you can also toggle advanced mode by pressing the check box. This will change question difficulty to that of the year above or an additional difficulty for users in year 6 with advanced mode. you can also update profile details (username and school year). Click 'Update Profile', fill in fields, then press 'Update'. If error message is present/update is unsuccessful try again in response to the error. You can also change your password. Click 'Change Password', fill in fields, then click second 'Change Password' button. If error message is present/update is unsuccessful try again in response to the error.

About us
About us page offers a paragraph about the app as well as admin contact details.

Faq
The FAQ page contains answers to common questions. Use the search bar to find specific key words. Click on any question to view its answer.

Running tests
open file in visual studio coad and run 'python -m http.server 8080'. Then in a new terminal while the other is running, run 'flutter pub get' then 'flutter test'.

Note- For compatability test drivers to work the corresponding version of the browser must be installed on the computer running the tests. Versions required:

```
For chrome: 135.x.xxxx.xx
For edge: 136.x.xxxx.xx
For firefox: 138.x.x(other versions of firefox may also work)
```

figure 0.7- Appendix F: Readme