Journal of Applied Research in Science and Humanities

New Technique of Mathematics Exams Based on Java and Dart

E.Mohamed, E.Nashed, M.Roshdy, M.Roshdy, R.Abdelsalam, S.Sayed, S.Elsayed

Supervisor: Dr. M.Hassan, Academic Degree: Associate professor, and Specialization: Statistics

Ain Shams University, Faculty of Education, Program Mathematics (English)

Abstract

This research project focuses on the development of a comprehensive application for managing online mathematics examinations using Java and Dart programming languages. We aim to explore novel methodologies in mathematics exam design, comparing traditional paper-based exams with online exam formats. With the evolution of digital education platforms, there is a growing need to explore innovative approaches to assessment that leverage the capabilities of modern programming languages. Leveraging the object-oriented capabilities of Java and the cross-platform development potential of Dart, our application aims to provide a user-friendly interface for both administrators and examinees. Key features include secure authentication, question randomization, real-time monitoring, and performance analytics. By employing robust encryption techniques and server-client communication protocols, we also aim to explore and compare the two programming languages, Java and Dart, starting from their history and reaching their uses and unique features. We also analyze how each of these two languages is used in developing applications and compare their capabilities and the development tools available to them, in addition to a comparison in performance, security, and flexibility. The efficacy of paper-based and online exams is evaluated through a comparative analysis of the student's performance. Additionally, educators's feedback and the parents of the students are taken into account to assess the feasibility and benefits of adopting online exam formats augmented by Java and Dart techniques.

Key Words: Java, Dart, Application, Mathematics Exam, programming language

1. Introduction:

In the ever-evolving landscape of education, the traditional methods of conducting mathematics exams are undergoing a transformative shift with the advent of innovative techniques rooted in technology. One such pioneering approach involves leveraging the power of Java and Dart programming languages to revolutionize the way mathematics exams are designed, administered,

and experienced. Traditionally, mathematics exams have often been static, paper-based assessments that measure students' knowledge through a series of predetermined questions. While these exams serve their purpose, they often lack the dynamism and interactivity necessary to fully engage students and accurately evaluate their understanding of mathematical concepts. By incorporating Java and Dart into mathematics exams, educators can create assessments that not only measure knowledge but also foster deeper understanding, critical thinking, and problem-solving skills. This innovative approach transcends the limitations of traditional exams, offering a more dynamic, interactive, and personalized learning experience for students.

About Java language: Java is one of the most popular programming languages software developers use worldwide. The goal of the project was to develop a programming language for consumer electronic devices, such as smart TVs, set-top boxes, and handheld controllers (Glushach, 2023). The language was initially called Oak after an oak tree that stood outside Gosling's office. Later, the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia (Murphy, 1996). Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar (Gosling, Holmes, & Arnold, 2005). Java was developed by James Gosling and others at Sun Microsystems as a class-based, object-oriented, and high-level programming language (Rahman, 2024). Sun Microsystems released the first public implementation as Java 1.0. It is designed to enable the" write once, run anywhere" (WORA) feature, which allows compiled Java code to run without further compilation on all platforms that support Java (Ikedilo, Osisikankwu, & Madubuike, 2015). Over the years, the language has been the foundation of millions of applications across many platforms, such as Windows, Macintosh, UNIX, Android-based handheld devices, Embedded Systems, and corporate solutions. Java operates on more than 3 billion smartphones, according to Oracle (which acquired Sun Microsystems) (Team, 2022). Java is an object-oriented, cross-platform programming language introduced by Sun Microsystems. Today, Java is fast, secure, and reliable. Developers use Java to construct applications in laptops, data centres, game consoles, scientific supercomputers, cell phones, and other devices. Java continued to evolve, with updates and new versions introducing features like the Swing GUI toolkit, the Collections Framework, and performance enhancements (Kumar, 2023). The evolution of Java is controlled through the Java Community Process (JCP), which allows users to propose additions and changes to the Java platform (O'Regan, 2018). One of the key features of Java is its platform independence because it is compiled into bytecode, which can run on any platform that supports the Java Virtual Machine (JVM). By this, the developers can write code once and run it on any platform without having to worry about platform-specific details (Martinez, Remegio, & Lincopinis, May 2023).

Because Java is a free-to-use and a versatile language, it builds localized and distributed software. Some common uses of Java include (Olumide, 2023):

- Web Application Development: Java is widely used in the development of web applications, including e-commerce sites, social media platforms, and enterprise-level systems. Java frameworks such as Spring, Struts, and Hibernate are popular choices for building web applications.

- Mobile Application Development: Java is the language of choice for building Android applications, which are used by billions of people worldwide. Java is also used in the development of cross-platform mobile applications using tools such as Xamarin and PhoneGap.
- Enterprise Application Development: Java is commonly used in the development of large-scale enterprise applications, such as banking and finance systems, healthcare systems, and supply chain management systems. Java frameworks such as JavaServer Faces (JSF), Java Persistence API (JPA), and Java Message Service (JMS) are popular choices for enterprise application development.
- Internet of Things (IoT): Java is increasingly being used in the development of IoT applications, which require lightweight and scalable software solutions. Java frameworks such as Eclipse IoT, Kura, and ThingWorx are popular choices for developing IoT applications.
- Artificial Intelligence (AI) and Machine Learning (ML): Java is being used in the development of AI and ML applications, particularly in the development of data analysis and processing tools. Java frameworks such as Apache Spark, Deeplearning4j, and H2O.ai are popular choices for developing AI and ML applications.
- Game Development: Java is used in the development of desktop and mobile games, with popular game engines such as LibGDX and jMonkeyEngine using Java as their primary programming language.

Features of Java Programming Language:

-Simple: Nobody wants to program in a language that makes life more difficult than it already is. James Gosling said that Java is "C++ without the knives, guns and clubs." C++, the language developed by Bjarne Stroustrup as an object-oriented extension to the C programming language, is very popular and powerful but has many features that, like weapons, are very dangerous in the wrong hands (perhaps any hands). One major simplification in Java is the way that memory is man- aged, using largely automatic processes rather than requiring the programmer to do this. Simplicity is, however, a relative term!

-Secure: As well as programmer error, programs are vulnerable to deliberate sabo security systems built into Java ensure that the code, once written, is not easy tamper with. This is particularly important for a language that is used to write grams that are distributed widely over networks (Parsons, 2012).

-High-performance: Java is faster compared to other interpreted programming languages because Java bytecode is more close to native code. It is still a little bit slower than a compiler-based language (e.g., C++). Java is an interpreted language, because of this it is slower than compiler-based languages, e.g., C, C++, etc.(Annigeri, 2021).

-Object-Oriented: All items in Java are displayed as objects. Anyone can structure software by combining diverse forms of entities that encompass both functions and data. OOP's incorporation in Java has contributed to its widespread use in the development of intricate software systems, particularly in enterprise settings where the properties of modularity and scalability are fundamental considerations.

- Architecture Neutral: Java is written to write once and run anywhere (WORA) feature. Java's architecture is neutral due to its ability to conceal the intricate details of the underlying system where Java code is executed. Several JRE implementations designed for various operating systems are available in Java.

-Portable: The reason behind Java being classified as portable is that you can compile code written in Java language, which results in the creation of a byte code. This byte code can be executed afterward with the help of a Java Virtual Machine. Java Virtual Machine is like an interpreter, which reads the compiled byte code and runs it. Why is Java more portable than C, C++? Because in C, C++ takes a lot of time (hours) to get compiled, while Java can run the byte code with the compiler. Since Java is a write once and run everywhere (WORA) feature.

-Dynamic: A Java program can dynamically change the resources it is using at runtime. This is useful in a distributed environment because it means that the program can be flexi- ble in terms of size and behavior. It is also easy to write programs that use many different objects because it is easy for them to locate each other at runtime, even when they are in different places.

-Robust: Java is considered a secure and resilient programming language due to its efficient memory management system. One of the benefits of this technology is its lack of pointers, which minimizes security vulnerabilities. Additionally, it employs automatic garbage collection through the Java Virtual Machine, which effectively eliminates unused objects in a Java application. All these points make Java robust.

About Dart language: Flutter is an open-source user interface software development kit brought to life by the wizards at Google. This robust tool allows developers like us to construct high-quality, cross-platform applications for users across a broad spectrum, including iOS, Android, Linux, Mac, and Windows, from a single codebase. Flutter was created to provide developers with a powerful and efficient way to build cross-platform applications for multiple operating systems, including Android development, iOS, web browsers, Linux, macOS, and Windows (Vikas, 2023). Flutter was founded by Mehul Nariyawala and Navneet Dalal and is based in California, United States. It was acquired by Google. It was first introduced in 2015 but gained significant attention and released its first commercial Flutter app (Google, 2017). Since then, it has experienced mainstream breakthroughs and has continued to evolve and improve in terms of stability and functionality. In 2017, a significant event reshaped the mobile app development industry – launching the first commercial Flutter app. This pivotal moment unfolded when VGV's founding team unleashed an application built using Flutter. Despite being a new player in the tech field at that time, Flutter displayed its prowess by enabling full development and launch on both iOS and Android platforms within three months only. Google launched Flutter's first stable release in 2018 and released the first version Flutter 1.0, which would run on iOS and Android devices but had limited Web support, lack of third party integrations and huge file size. These problems paved the way for the release of Flutter 2.0 in 2021. Flutter 2.0 allowed developers to develop apps on Android, Desktop and Web with the same codebase. Besides, it introduced Web Support across all platforms, Desktop support, null safety and integration of Google Ads in your app. While these updates were praised and gave way to over 450,000 Flutter apps on the Play Store, there still remained a need for Flutter to evolve from a mobile centric to a multiplatform tool. In 2022, the release of Flutter 3.0 and Dart 2.17 was announced simultaneously (Nasir, 2023). Flutter nowadays has steadily grown and provided possibilities not only for iOS and Android mobile development but also for web and desktop

applications as well. Flutter framework makes it easy for you to build user interfaces that are beautiful, fast, and responsive. The framework is written using C, C++, and Dart languages, and uses Google's skia Graphics Engine for user interface rendering. The framework is also extensible, so you can easily add new features and functionality. Flutter offers good quality for reasonable costs, as well as great usability and speed. Development goes faster when the same code is used for both iOS and Android applications. Flutter's singular codebase speeds up time-to-market while cutting mobile app development costs significantly. Flutter app performance is equivalent to that of native real-time applications. Unlike other frameworks, Flutter apps do not need a bridge to interact with native components. Since these bridges typically cause performance issues, this gives Flutter a decided advantage. Flutter "hot reload" feature allows developers to change the code on emulators, simulators, or real devices, and see results in real-time. While Flutter app development presents a lot of positives, it is also important to consider the less positive aspects of the framework: Dart is not very popular. While it is a great programming language, developers are often much more likely to consider languages such as Java or Kotlin. Some components are only available for iOS or Android, but not both. These types of components more often support Android since Flutter comes from Google, and Android developers are typically more interested in Flutter than iOS developers. As you venture into the world of Flutter app development, it's essential to familiarize yourself with the Dart programming language. Dart serves as the foundation for building Flutter applications (lund, 2011), and understanding its fundamentals is crucial for writing clean, efficient code. Dart is a programming language for fast apps on any platform designed by Lars Bak and Kasper Lund. Dart was unveiled at the (GOTO, 2011) conference in Aarhus, Denmark. Dart 1.0 was released (Lars, 2013). Dart had a mixed reception at first. Some criticized the Dart initiative for fragmenting the web because of plans to include a Dart VM in Chrome. Those plans were dropped with the Dart 1.9 release. Focus changed to compiling Dart code to JavaScript (Seth, 2019). Dart 2.0 was released with language changes including a type system (Kevin, 2018). Dart 2.6 introduced a new extension, dart2native. This extended native compilation to the Linux, macOS, and Windows desktop platforms. Earlier developers could create new tools using only Android or iOS devices with this extension, developers could deploy a program into self-contained executables. The Dart SDK doesn't need to be installed to run these self contained executables. The Flutter toolkit integrates Dart, so it can compile on small services like backend support (Paul, 2019). Dart 3.0 was released in 2023 with changes to the type system to require sound null safety. This release included new features like records, patterns, and class modifiers. Dart 3 also previewed support for Web Assembly (Michael, 2023). Dart is an object-oriented, class-based, garbage-collected language, developed by Google using the C-style syntax. Google introduced Dart as a general-purpose programming language designed to address the shortcomings of JavaScript in large-scale application development. It was introduced to provide a productive and efficient language for building high-quality applications. Dart combines the best features of various programming languages to offer a powerful and flexible development experience. It is designed to be easy to learn and use for building web, mobile, and desktop applications. It supports most of the common concepts of programming languages like

classes, interfaces, and functions, unlike other programming languages. Dart language does not support arrays directly. It supports collection, which is used to replicate the data structure such as arrays, generics, and optional typing. Dart compilers are a key part of any Dart developer's toolkit, translating human-readable Dart code into machine code or JavaScript. Dart compilers come in different Flavors, each designed for a specific use case. The two main types are the Dart-to-JavaScript compiler (dart2js) and the Dart Ahead-of-Time (AOT) compiler. The Dart-to-JavaScript compiler, or dart2js, is used when you want to run your Dart code in a web browser. Since browsers can't natively understand Dart, the code needs to be compiled into JavaScript, a language that browsers can interpret. The Dart AOT compiler, on the other hand, is used for compiling Dart code into machine code. This is particularly useful when you're building mobile applications with Flutter, Dart's UI toolkit. AOT compilation results in fast startup and predictable performance, without the need for a JavaScript bridge. In addition to AOT compilation, Dart also supports Just-In-Time (JIT) compilation. This is a type of compilation where the Dart code is compiled "on the fly" during the execution of the program. JIT compilation allows for hot reloading, where changes in the code can be injected into a running application. It's a feature that significantly boosts productivity during development.

Difference between Java and Dart

Dart and Java are both popular programming languages, but they have some significant differences. Here is a comparison of Dart and Java to help you decide which language is best for your needs.

Java and Dart Similarities

Both are object-oriented programming languages that support classes, interfaces, inheritance, and more.
They are both based on and follow C-style syntax, with curly brackets for blocks, variables declared as types, and semicolons to end statements.
Dart and Java both support static typing, although Dart provides it as an option, whereas Java makes it mandatory.
Irrespective of what language you use—Java or Dart—they use garbage collection for automatically managing memory allocation and deallocation.
You can build event-driven apps with either programming languages (Dart or Java).
Both have a rich list of standard libraries and frameworks for everyday development needs.
Supported Platforms: **Java Uses Time Compilation to Improve Start-up Time**.

JVM to help Windows, Linux, and Android, whereas
Dart runs on native code using Flutter's cross-platform framework.
Null Safety: Java is prone to Null Pointer Exceptions, whereas Dart has robust built-in null-safety features.
Concurrency: Java uses threads and locks, whereas Dart uses isolate and message passing to achieve coexistence

Java Static Typing vs Dart Optional Typing

Java is statically typed, which means the program expects the variables to be declared before

they are assigned values. This allows the compiler to decide whether the given variable should be allowed to perform the actions it requests. This enables type safety during compilation, preventing programs from accessing memory inappropriately. On the other hand, Dart is optionally typed, which means you can choose between static typing, like Java, and dynamic typing. Choosing between the two enables you to benefit from the type safety features of static typing and/or the flexibility of dynamic typing, where the compiler or interpreter assigns a type to all variables at runtime

Java Performance Strengths

Java has many built-in features and capabilities, making it a highly performant programming language.
language that powers Some of the most complex, scalable, and data-heavy enterprise applications. Here are the performance benefits of using Java over Dart.
Just-in-Time Compiler: Java compiles applications into bytecode, which gets executed by the Java Virtual Machine. The JVM uses the JIT compiler to convert this bytecode into native machine code during runtime, which results in highly optimized performance.
Multithreading: Java also provides robust support for multithreading, as we already saw, which makes it capable of building scalable and concurrent applications.

Mature and Optimized Runtime: Java has been around since the 1990s. It has continuously evolved and been worked upon by many developers to optimize it and improve its performance capabilities. Hence, it has a well-performing runtime environment.

Dart Performance Strengths

Ahead of Time Compilation: Dart follows AoT compilation, which means its code can be compiled ahead of time to native machine code for targeted platforms like web browsers and mobile devices. This helps create highly performant mobile applications and web applications.

Frontend Optimization: Dart combined with the Flutter framework for mobile app development services can result in highly responsive and performing user interfaces and animations.

Table1. Feature of Java and Dart.

Feature	**Dart**	**Java**
Purpose	Primarily used for building mobile and web applications	Used for a wide range of applications, including Android mobile apps, web applications, and enterprise systems

Syntax and Structure	More concise and expressive, similar to other modern programming languages	More verbose syntax, familiar to programmers with a background in languages like C or C++
Typing	Dynamically-typed	Statically-typed
Performance	Fast execution times	Fast execution times, but maybe faster and more efficient due to static typing
Developer Community	Niche language with a smaller developer community	Extremely popular language with a large and active developer community
Tools and Frameworks	May not have the same level of support and resources as Java	A wide range of tools and frameworks are available

The choice between Dart and Java will depend on your specific needs and the type of project you are working on. If you are building a mobile or web application and want concise and expressive language with fast execution times, then Dart may be the right choice. On the other hand, if you are working on a larger-scale project that requires a more established and widely supported language, then Java may be the better option.

Dart and Java are both powerful programming languages with their own strengths and weaknesses. Java is a mature, widely-used language that is well-suited for building large-scale enterprise applications. Dart, on the other hand, is a newer language that is designed to be easy to learn and use, making it a good choice for building web and mobile applications.

2.New Technique of Mathematics Exam Based on Java.

In this section, we present a new technique for mathematics exams based on Java:
For the beginning, we exam the student by paper, and the teacher checks it by himself as usual, but now there are so many ways to test the students, such as online exams, so we build an offline app based on Java that is easy to use for each of the teacher and student So we will talk about this app. Our app is divided into two parts, one for the teacher to make the exam and another for the student to solve it.

In the first part, the teacher enters login information (username = admin and password = admin). Then he can create more than one topic, add questions for each topic to test the students, and delete specific topics from the topics so the students will not see them, and he can also create (add or delete) more than one note to remind the student of some information before the exam, and the teacher can view the student's results easily without effort, and finally, logout from the app.

In the second part, the student can enter the app by logging in (his name and password), if he does not log in, he can sign up (register).The student can choose a specific topic that the teacher also chose for the exam, and he can see a note before the quiz. When the student enters the topic, he should answer the questions (choose the correct answer) before the end of timer, then view the result for the student at the end of the exam to know his level in the exam.

At the end, all the features. built by Android Studio using the Java language to make it easy to use and save effort and time.

For coding, see Appendix A

Figure 1 shows the flowchart of our application MCQ App, and Figure 2 shows the steps to use this application.

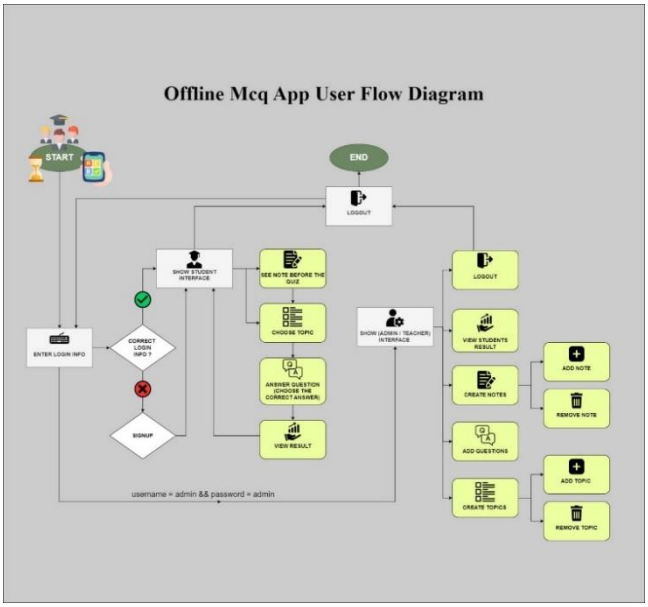


Fig 1. Flowchart of MCQ App

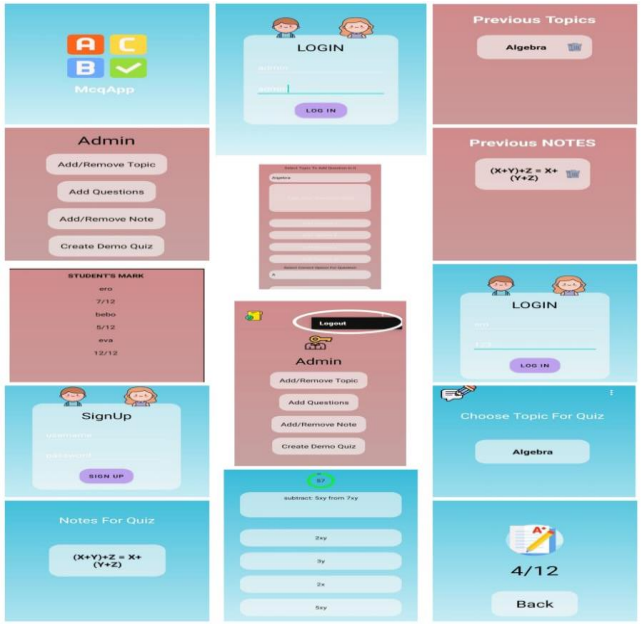


Fig 2. the steps of MCQ App

3.New Technique of Mathematics Exam Based on Dart.

After we talk about app built by Java, let's talk about another app built by dart. Our application is divided into two parts: the first is for the teacher to take the test, and the second is for the student to solve it. In the first part, the teacher registers his own email and must start (dr.) For example,

(dr.salmaelsayed@gmail.com) to put the test in, he presses sign up, and he goes to the Google Form to put the test in. He can put up more than one test and an unlimited number of questions (the questions are multiple choice), and the teacher determines the answers and test scores.

This part is not seen by the student who passes the test, and the teacher can view students' grades and also know how many people attended the test, then take the test link and put it in the URL section, inform the students of the test date, and then log out of the application. In the second part, the student can access the application by logging in with his own email. The test link appears, and the student enters and answers the test (multiple choice). After he finishes the solution, the result is shown to him, with the answer corrected if there is an error in the solution. And finally, all the features. It was designed by Android Studio using the Dart language to make it easier to use and save effort and time.

For coding, see Appendix B

Figure 3 shows the flowchart of our application Teaching project, and Figure 4 shows the steps to use this application.

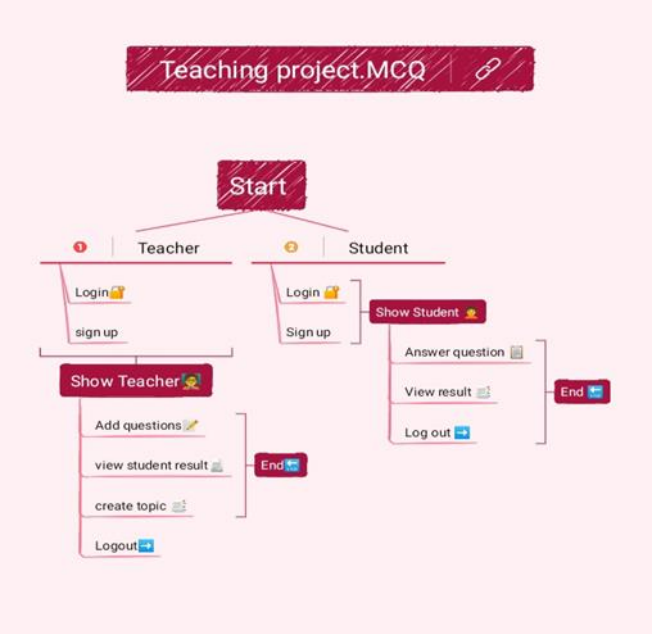


Fig 3. Flowchart of Teaching project

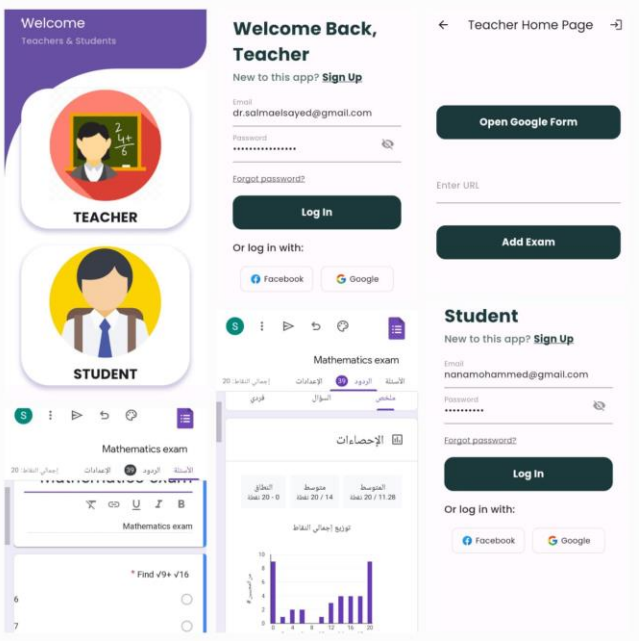


Fig 4. The steps of Teaching project

4.Methods of Research and the tools used

This section of the research paper will describe a questionnaire about electronic quizzes compared to traditional paper-and-pencil quizzes that aims to collect opinions, experiences and regarding of

students, teachers, and parents about traditional paper-based examinations and electronic examinations. This questionnaire includes questions about personal preferences, challenges faced by participants, and advantages and disadvantages of each type of exam. It aims to Identify the strong and weak points in both paper and electronic exams and understand individuals' preferences and uses of technology in educational assessments.

Objectives:

- Investigate student preference for electronic quizzes vs. traditional quizzes.
- Identify perceived advantages and disadvantages of each format.
- Assess the impact of quiz format on student engagement and learning.

Questionnaire Design:

- The questionnaire consisted of 35 questions that covered key issues around mathematics electronic quizzes and traditional quizzes. This questionnaire addresses three categories (students, teachers, and parents). The questionnaire will be divided into three sections:

- The first section asks students about their recent experiences with both electronic and traditional quizzes.
- The second section asks the teacher about their recent experiences with both electronic and traditional quizzes.
- The Third section asks parents about their children's experiences with both electronic and traditional quizzes.

For more details about questionnaire see Appendix C

5. Results of Research

The questionnaire was administered online to 30 Students, 35 teachers and 30 parents. 88 completed the questionnaire, resulting in a response rate of 92.6%. The results of questionnaire responses data are provided in these charts.

This figure show the student's response.

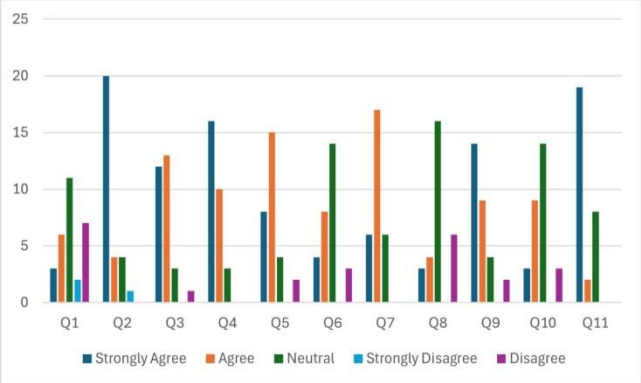


Fig 5. Students' respond.

When asked students about enhancing math problem-solving skills, 65.5 % of respondents replied (Strongly agree). Electronic quizzes are more engaging and interactive than traditional quizzes 69 % of respondents replied (Strongly agree).

This figure shows the teacher's response.

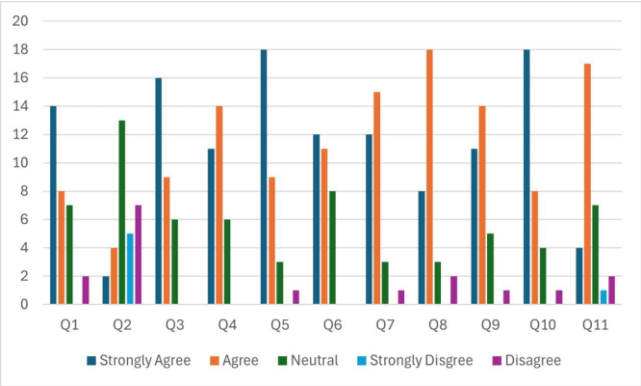


Fig 6. Teacher's respond.

When we asked the teacher about the ease of preparation of the electronic quizzes, 45.2% of response replied (Strongly agree). Electronic quizzes are a valuable tool for assessing student learning, 51.6 % of respondents replied (Strongly agree).

This figure shows the parent's response.

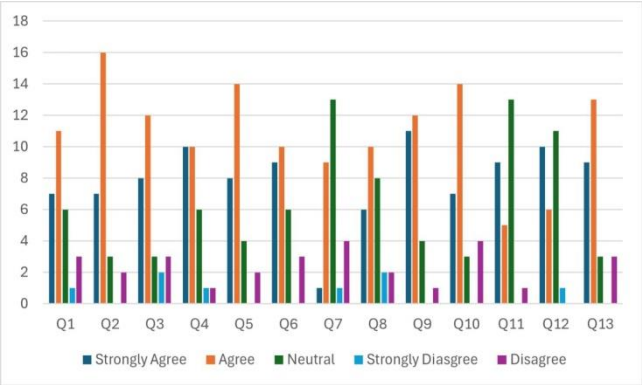


Fig 7. Parents respond.

When we asked the parent if the electronic quizzes motivate the children to learn and improve,35.7% of respondents replied (Strongly agree). I am interested in receiving feedback from the teacher about my child's performance on electronic quizzes, 39.3% of respondents replied (Strongly agree).

Overall, most of the participants stated that they prefer online quizzes and only a small proportion indicate traditional quizzes.

In conclusion, this research has improved our current understanding of electronic quizzes for as many different categories as possible (student, teacher, and parents).

6. Conclusion

In this paper, we introduced a brief for Java and Dart and concluded that Dart and Java are both powerful programming languages with their strengths and weaknesses. Java is a mature, widely-used language that is well-suited for building large-scale enterprise applications. Dart, on the other hand, is a newer language that is designed to be easy to learn and use, making it a good choice .for building web and mobile applications
We designed two applications using Java and Dart programming languages to apply for online mathematics exams. We compare between online exams and paper exams. Furthermore, we have applied a questionnaire for students, parents and teachers and discovered that online exams are much more efficient for both the teachers and the learners compared to paper exams. For instance, the majority of students found online tests more interactive and effective and teachers see that online tests are valuable tools for assessing student learning despite some challenges in preparing them. Parents were interested in feedback from teachers about their children's performance on these tests. The results indicate a general trend towards preferring electronic tests over traditional tests. Additionally, the teacher only needs to input the questions once then the questions will automatically be displayed to the learners. While learners can experience a more challenging environment and receive instant results after submitting the exam.

7. Future Work

There are many features in the future you could add to a quiz application, depending on the specific needs of your users.
First App, MCQ App
Here are a few features to consider:
1) We look forward to making the application available on all mobile phones for all students and teachers.

2) Randomized questions: Give users the option to randomize the order of the questions and answer choices. This can help prevent cheating and make the quizzes more challenging.
3) Leaderboards: Allow users to compete with each other by displaying a leaderboard of top scores. This can motivate users to take more quizzes and try to improve their scores.

Second App, Teaching project
Here are a few features to consider:
1. Add a special section for review before the test.
2. Create a special section for parents to know their children's grades after each test

Acknowledgement

At the end, we would like to express our deep thanks to Dr. Marwa for making great effort and dedication to this project.
"A big thanks to Eng. Irini Ashraf, who graduated from computer science at ASU, for her invaluable contribution. Your efforts were instrumental in achieving this project and app development".
"We appreciate your great efforts on the project. Thank you, Eng. Mohamed Mostafa."

References and Sources

Annigeri, S. (2021). Top 10+ Features Of Java Programming Language You Must Know. Retrieved from CS Electrical and Electronics: https://cselectricalandelectronics.com/top-10-features-of-java-programming-language-you-must-know/

Glushach, R. (2023). The Evolution of Java: A Historical Perspective. Retrieved from medium: https://romanglushach.medium.com/the-evolution-of-java-a-historical-perspective-e15c3d7e5f85

Gosling, J., Holmes, D. C., & Arnold, K. (2005). The Java programming language.

Ikedilo, O., Osisikankwu, P., & Madubuike, C. (2015). A Critical Evaluation of Java as a Good Choice for Introductory Course. International Journal of Research, 02(12), 847-853.

Kumar, P. (2023). Java Evolution: Navigating Major Updates. Retrieved from medium: https://medium.com/@parimalkumar261/exploring-java-version-changes-a-journey-through-major-updates-4859c92e44b7

Martinez, D., Remegio, A., & Lincopinis, D. (2023). A Review on Java Programming Language.

Murphy, K. (1996). So why did they decide to call it Java? Retrieved from InfoWorld: https://www.infoworld.com/article/2077265/so-why-did-they-decide-to-call-it-java.html

O'Regan, G. (2018). The innovation in computing companion. Springer.

Olumide, S. (2023). What is Java Used For in 2023? The Java Programming Language and Java Platform Strengths. Retrieved from freecodecamp: https://www.freecodecamp.org/news/what-is-java-used-for/

Parsons, D. (2012). Foundational Java: Key Elements and Practical Programming. Springer Science & Business Media.

Rahman, S. A. (2024). History of Java: From Island Origins to Programming Powerhouse. Retrieved from shiksha online:

https://www.shiksha.com/online-courses/articles/history-of-java/

Team, U. E. (2022). History of Java Programming Language| UNext. Retrieved from u-next: https://u-next.com/blogs/java/history-of-java/

Google. (2010). wikipedia. Retrieved from https://en.m.wikipedia.org/wiki/Firebase#:~:text=User%20privacy%20controversies-,History,chat%20functionality%20into%20their%20websites.: https://firebase.google.com/docs/auth/flutter/start

Google. (2011). what is dart. Retrieved from http://radar.oreilly.com/2012/03/what-is-dart.html

Google. (2017). wikipedia. Retrieved from https://en.m.wikipedia.org/wiki/Flutter_(software): https://flutter.dev/

Google. (2019). Dart 2.6 released with dart2native . Retrieved from https://sdtimes.com/goog/dart-2-6-released-with-dart2native/

Google. (2023). Class modifiers . Retrieved from https://dart.dev/language/class-modifiers

Google. (2023). Dart language evolution . Retrieved from https://dart.dev/guides/language/evolution

Google. (2024). Dart language evolution . Retrieved from https://dart.dev/guides/language/evolution

Google, J. B. (2013). wikipedia. Retrieved from https://en.m.wikipedia.org/wiki/Android_Studio: https://developer.android.com/studio?gad_source=1&gclid=CjwKCAiA0PuuBhBsEiwAS7fsNXvUmhQVwOwWfoR-y-kvmqe3UJcl41b3B8U8xVbQjTFZSeLzeBsRPoNRo-CaIkQAvD_BwE&gclsrc=aw.ds

GOTO, c. (2011). Dart,a new programming language. Retrieved from http://goto-con.com/aarhus-2011/presentation/Opening%20Keynote:%20Dart,%20a%20new%20programming%20language%20for%20structured%20web%20programming

Kevin, M. (2018). Announcing Dart2 Stable. Retrieved from https://medium.com/dartlang/dart-2-stable-and-the-dart-web-platform-3775d5f8eac7

Lars, B. (2013). Dart 1.0:A stable SDK for structured web apps. Retrieved from https://news.dartlang.org/2013/11/dart-10-stable-sdk-for-structured-web.html

lund, B. l. (2011). wikipedia. Retrieved from https://en.m.wikipedia.org/wiki/Dart_(programming_language): https://dart.dev/

Michael, T. (2023). Announcing Dart3. Retrieved from https://medium.com/dartlang/announcing-dart-3-53f065a10635

Nasir, Z. (2023). What is Flutter. Retrieved from https://www.walturn.com/insights/four-years-of-flutter

Paul, k. (2019). Dart 2.5 brings native compilation to the desktop. Retrieved from https://www.infoworld.com/article/3454623/dart-26-brings-native-compilation-to-the-desktop.html

Seth, L. (2019). Dart News and Updates. Retrieved from http://news.dartlang.org/2015/03/dart-for-entire-web.html

Vikas, K. (2023). History of Flutter. Retrieved from https://www.techahead-corp.com/knowledge-center/history-of-flutter/

Google, T. o. (n.d.). Retrieved may 16, 2013

Android - SQLite Database. (n.d.). Retrieved from tutorialspoint: https://www.tutorialspoint.com/android/android_sqlite_database.htm

Android Sqlite tutorial. (n.d.). Retrieved from Javatpoint: https://www.javatpoint.com/android-sqlite-tutorial#:~:text=SQLite%20is%20an%20open%2Dsource,database%20setup%20or%20administration%20task.

Build software better, together. (n.d.). Retrieved from github: https://github.com/topics/quiz-app

Build software better, together. (n.d.). Retrieved from github: https://github.com/topics/quiz-app-android

Download Android Studio & App Tools. (n.d.). Retrieved from developer.android: https://developer.android.com/studio?gad_source=1&gclid=CjwKCAiAuN-GuBhAkEi-wAGId4asdgPcLOZ2AZfEcCX5v2zWAsgvpjwp4mFAftG3u-UPh_npLt_gmvrRoCX0oQAvD_BwE&gclsrc=aw.ds

Download the latest Java LTS free. (n.d.). Retrieved from oracle: https://www.oracle.com/java/technologies/downloads/

How to get items from recycler view adapter. (n.d.). Retrieved from stackoverflow: https://stackoverflow.com/questions/59838144/how-to-get-items-from-recycler-view-adapter

lmf. (2023). Build Your First Android App in Java. Retrieved from developer.android: https://developer.android.com/codelabs/build-your-first-android-app#0

Progress indicator. (n.d.). Retrieved from developer.android: https://developer.android.com/design/ui/wear/guides/components/progress-indicator

Sinicki, A. (2021). How to call a method in Java. Retrieved from androidauthority: https://www.androidauthority.com/how-to-call-a-method-in-java-1149853/

Splash screens. (n.d.). Retrieved from developer.android: https://developer.android.com/develop/ui/views/launch/splash-screen

Appendix A

Appendix

Contents

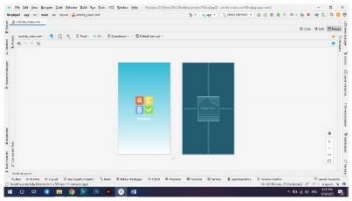
1

2

MAIN ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.CountDownTimer;

public class SplashScreenActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //we will check later if user is already logged in or
not.

        new CountDownTimer(3000,1000){
            @Override
            public void onTick(long l) {

            }

            @Override
            public void onFinish() {
                SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONER",MODE_PRIVATE);

if(sharedPreferences.getString("USER","").isEmpty()){
                    startActivity(new
Intent(SplashScreenActivity.this,LoginActivity.class));
                }else{
                    startActivity(new
Intent(SplashScreenActivity.this,TopicActivity.class));
                }
                SplashScreenActivity.this.finish();
            }
        }.start();

    }
}
```

3



LOGIN ACTIVITY CODE

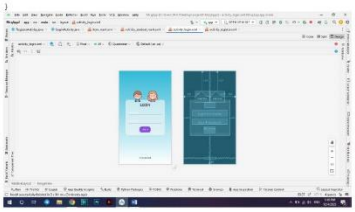
```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Date;

public class LoginActivity extends AppCompatActivity {


    EditText userNameEditText;
    EditText passwordEditText;
    MyDbHelper dbHelper;
```

4

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        dbHelper = new MyDbHelper(this);
        userNameEditText = findViewById(R.id.loginUsername);
        passwordEditText = findViewById(R.id.loginPassword);

        findViewById(R.id.loginBtn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

loginUser(userNameEditText.getText().toString().trim(),passwordE
ditText.getText().toString().trim());
            }
        });


findViewById(R.id.loginCreateAccountLink).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(LoginActivity.this,RegisterActivity.class));
                LoginActivity.this.finish();
            }
        });

    }


    public void loginUser(String username,String password){
        if(username.isEmpty() || password.isEmpty()){
            Toast.makeText(this, "Please fill both the fields!",
Toast.LENGTH_SHORT).show();
        }else if(username.equalsIgnoreCase("admin") &&
password.equalsIgnoreCase("admin")){
            startActivity(new
Intent(LoginActivity.this,AdminMainActivity.class));
            LoginActivity.this.finish();
        }else{
            int result = dbHelper.loginUser(username, password);
            if(result == -1){
                Toast.makeText(this, "Login failed!",
Toast.LENGTH_SHORT).show();
```

5

```
            }else{
                Toast.makeText(this, "Welcome!",
Toast.LENGTH_SHORT).show();
                SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONER",MODE_PRIVATE);
                SharedPreferences.Editor editor =
sharedPreferences.edit();
                editor.putString("USER",username);
                editor.apply();
                startActivity(new
Intent(LoginActivity.this,TopicActivity.class));
                LoginActivity.this.finish();
            }
        }
    }
```



SIGNUP ACTIVITY CODE

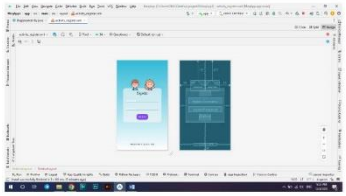
```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
```

6

```
import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
public class RegisterActivity extends AppCompatActivity {

    EditText userNameEditText;
    EditText passwordEditText;
    MyDbHelper dbHelper;
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        dbHelper = new MyDbHelper(this);
        userNameEditText = findViewById(R.id.registerUsername);
        passwordEditText = findViewById(R.id.registerPassword);
        findViewById(R.id.registerSignUpBtn).setOnClickListener
                (view ->
registerUser(userNameEditText.getText().toString().trim()
,passwordEditText.getText().toString().trim()));


        findViewById(R.id.AccountLink).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(RegisterActivity.this,LoginActivity.class));
                RegisterActivity.this.finish();
            }
        });
    }

    public void registerUser(String username,String password){
        if(username.isEmpty() || password.isEmpty()){
            Toast.makeText(this, "Please fill both the fields!",
Toast.LENGTH_SHORT).show();
        }else if(username.equalsIgnoreCase("admin")){
            Toast.makeText(this, "Please choose another
username!", Toast.LENGTH_SHORT).show();
        }else {
            long result = dbHelper.createUserAccount(username,
```

7

```
password);
            if(result == -1){
                Toast.makeText(this, "Error!",
Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(this, "Welcome!",
Toast.LENGTH_SHORT).show();
                SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONER",MODE_PRIVATE);
                SharedPreferences.Editor editor =
sharedPreferences.edit();
                editor.putString("USER",username);
                editor.apply();
                startActivity(new
Intent(RegisterActivity.this,TopicActivity.class));
                RegisterActivity.this.finish();
            }
        }
    }
}
```



TOPIC ACTIVITY CODE

8

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.PopupMenu;

import java.util.ArrayList;

public class TopicActivity extends AppCompatActivity {

    RecyclerView topicsRecyclerView;
    TopicsAdapter topicsAdapter;
    MyDbHelper dbHelper;

    ImageView img;
    ArrayList<String> topicsList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_topic);

        topicsRecyclerView =
findViewById(R.id.topics_recyclerView);


        dbHelper = new MyDbHelper(this);
        topicsList = new ArrayList<>();
        topicsList = dbHelper.getAllTopics();

        topicsAdapter = new TopicsAdapter(topicsList, this, new
MyOnClickListener() {
            @Override
            public void onClick(int position) {
                Intent intent = new
Intent(TopicActivity.this,QuestionActivity.class);
                intent.setAction(topicsList.get(position));
                startActivity(intent);
            }
```

9

```
        }));

        topicsRecyclerView.setAdapter(topicsAdapter);
        topicsRecyclerView.setLayoutManager(new
LinearLayoutManager(this));

        findViewById(R.id.topic_more).setOnClickListener(view ->
showPopUpMenu(view));

        img = findViewById(R.id.revisionimg);
        img.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(TopicActivity.this,SummaryActivity.class));
                TopicActivity.this.finish();
            }
        });

    }

    private void showPopUpMenu(View view){
        PopupMenu popupMenu = new
PopupMenu(TopicActivity.this,view);
        MenuInflater inflater = popupMenu.getMenuInflater();

inflater.inflate(R.menu.topic_screen_menu,popupMenu.getMenu());
        popupMenu.show();
        popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONS",MODE_PRIVATE);
                SharedPreferences.Editor editor =
sharedPreferences.edit();
                editor.putString("USER","");
                editor.apply();
                startActivity(new
Intent(TopicActivity.this,LoginActivity.class));
                TopicActivity.this.finish();
                return true;
            }
```

10

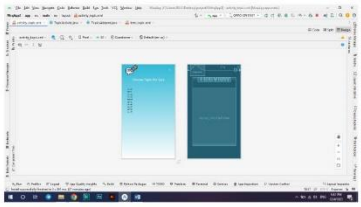
```
        });
    }
}
```

TOPICS ADAPTER :

```
package com.example.McqApp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class TopicsAdapter extends
RecyclerView.Adapter<TopicsAdapter.MyViewHolderNew> {

    ArrayList<String> topics;
    Context context;
    MyOnClickListener onClickListener;

    public TopicsAdapter(ArrayList<String> topics, Context
context, MyOnClickListener onClickListener) {
        this.topics = topics;
        this.context = context;
        this.onClickListener = onClickListener;
    }

    @NonNull
    @Override
    public MyViewHolderNew onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        return new
MyViewHolderNew(LayoutInflater.from(context).inflate(R.layout.it
em_topic,parent,false));
    }

    @Override
    public int getItemCount() {
        return topics.size();
    }
```

11

```
    @Override
    public void onBindViewHolder(@NonNull MyViewHolderNew
holder, int position) {

holder.topic.setText(topics.get(holder.getAdapterPosition()));
        holder.topic.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

onClickListener.onClick(holder.getAdapterPosition());
            }
        });
    }

    protected static class MyViewHolderNew extends
RecyclerView.ViewHolder {

        TextView topic;

        public MyViewHolderNew(@NonNull View itemView) {
            super(itemView);
            topic = itemView.findViewById(R.id.topicTextView);
        }
    }
}
```



12

NOTE/SUMMARY ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.TextView;

import java.util.ArrayList;

public class SummaryActivity extends AppCompatActivity {
    ImageView img;

    RecyclerView notesRecyclerView;
    NotesAdapter notesAdapter;
    MyDbHelper dbHelper;
    ArrayList<String> noteslist;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_summary);

        img = findViewById(R.id.goPng);
        img.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(SummaryActivity.this,TopicActivity.class));
                SummaryActivity.this.finish();
            }
        });
```

13

```
        notesRecyclerView =
findViewById(R.id.notes_recyclerView);

        dbHelper = new MyDbHelper(this);
        noteslist = new ArrayList<>();
        noteslist = dbHelper.getAllSummaries();

        notesAdapter = new NotesAdapter(noteslist, this, new
MyOnClickListener() {
            @Override
            public void onClick(int position) {
                Intent intent = new
Intent(SummaryActivity.this,TopicActivity.class);
                intent.setAction(noteslist.get(position));
                startActivity(intent);
            }
        });

        notesRecyclerView.setAdapter(notesAdapter);
        notesRecyclerView.setLayoutManager(new
LinearLayoutManager(this));

    }

}
```

NOTES ADAPTYER :

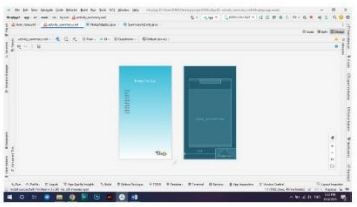
```
package com.example.McqApp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
```

14

```
public class NotesAdapter extends
RecyclerView.Adapter<NotesAdapter.MyViewHolderNew> {

    ArrayList<String> notes;
    Context context;
    MyOnClickListener onClickListener;

    public NotesAdapter(ArrayList<String> notes, Context
context, MyOnClickListener onClickListener) {
        this.notes = notes;
        this.context = context;
        this.onClickListener = onClickListener;
    }

    public NotesAdapter.MyViewHolderNew
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new
MyViewHolderNew(LayoutInflater.from(context).inflate(R.layout.it
em_note,parent,false));
    }

    public int getItemCount() {
        return notes.size();
    }

    public void onBindViewHolder(@NonNull
NotesAdapter.MyViewHolderNew holder, int position) {

holder.note.setText(notes.get(holder.getAdapterPosition()));
        holder.note.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

onClickListener.onClick(holder.getAdapterPosition());
            }
        });
    }

    protected static class MyViewHolderNew extends
RecyclerView.ViewHolder {

        TextView note;
        public MyViewHolderNew(@NonNull View itemView) {
            super(itemView);
            note = itemView.findViewById(R.id.noteTextView);
```

15

```
        }
    }
}
```



QUESTION ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.animation.AnimatorInflater;
import android.animation.ObjectAnimator;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import
com.google.android.material.progressindicator.CircularProgressIn
dicator;
```

16

```
import com.google.android.material.snackbar.Snackbar;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class QuestionActivity extends AppCompatActivity {

    int backCount = 0, questionCount = 0 , correctAnswers = 0;

    boolean stopResponse = true;

    MyDbHelper myDbHelper;
    ArrayList<Question> questionArrayList;
    CircularProgressIndicator progressIndicator;
    TextView progressText, optionA, optionB, optionC, optionD,
questionTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_question);
        myDbHelper = new MyDbHelper(this);
        questionArrayList = new ArrayList<> ();
        questionArrayList =
myDbHelper.getQuestions(getIntent().getAction());
        progressIndicator = findViewById(R.id.progressBar);

        questionTextView = findViewById(R.id.questionDisplay);
        optionA = findViewById(R.id.optionA);
        optionB = findViewById(R.id.optionB);
        optionC = findViewById(R.id.optionC);
        optionD = findViewById(R.id.optionD);

        progressText = findViewById(R.id.progressTextView);

        if(questionArrayList.isEmpty()){
            questionTextView.setVisibility(View.INVISIBLE);
            optionA.setVisibility(View.INVISIBLE);
            optionB.setVisibility(View.INVISIBLE);
            optionC.setVisibility(View.INVISIBLE);
            optionD.setVisibility(View.INVISIBLE);

            View parentLayout =
findViewById(android.R.id.content);
            Snackbar.make(parentLayout, "No Questions
Available!", Snackbar.LENGTH_INDEFINITE)
```

17

```
                .setAction("CLOSE", new
View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        onBackPressed();
                        onBackPressed();
                    }
                })

.setActionTextColor(getResources().getColor(android.R.color.holo
red_light,getTheme()))
                .show();
        }else{
            loadQuestion();

            ObjectAnimator animator = (ObjectAnimator)
AnimatorInflater.loadAnimator(this,R.animator.flip);
            animator.setDuration(500);

            optionA.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    if(stopResponse){

if(questionArrayList.get(questionCount).getCorrectOption().equal
sIgnoreCase("A")){
                            correctAnswers++;
                            animator.setTarget(view);
                            animator.start();
                        }else {
                            view.setBackgroundColor(Color.RED);
                        }
                        stopResponse = false;
                    }
                }
            });

            optionB.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    if (stopResponse) {
                        if
(questionArrayList.get(questionCount).getCorrectOption().equalsI
gnoreCase("B")) {
                            correctAnswers++;
```

18

```
                        animator.setTarget(view);
                        animator.start();
                    } else {
                        view.setBackgroundColor(Color.RED);
                    }
                }
                stopResponse = false;
            }
        });

        optionC.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (stopResponse) {
                    if
(questionArrayList.get(questionCount).getCorrectOption().equalsI
gnoreCase("C")) {
                        correctAnswers++;
                        animator.setTarget(view);
                        animator.start();
                    } else {
                        view.setBackgroundColor(Color.RED);
                    }
                }
                stopResponse = false;
            }
        });

        optionD.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (stopResponse) {
                    if
(questionArrayList.get(questionCount).getCorrectOption().equalsI
gnoreCase("D")) {
                        correctAnswers++;
                        animator.setTarget(view);
                        animator.start();
                    } else {
                        view.setBackgroundColor(Color.RED);
                    }
                }
                stopResponse = false;
            }
        });
```

19

```
        }

    }

    protected void loadQuestion(){
        if(questionCount != questionArrayList.size()){
            stopResponse = true;
            startTimer(questionCount);
        }else{
            addResultToDatabase();
            Intent intent = new
Intent(QuestionActivity.this,ResultActivity.class);

intent.setAction(correctAnswers+"/"+questionArrayList.size());
            startActivity(intent);
            QuestionActivity.this.finish();
        }
    }


    private void addResultToDatabase() {
        SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONER",MODE_PRIVATE);
        SimpleDateFormat dateFormat = new
SimpleDateFormat("dd/MM/yy");

myDbHelper.addResult(sharedPreferences.getString("USER",""),date
Format.format(new
Date()),correctAnswers+"/"+questionArrayList.size(),getIntent().
getAction());
    }

    private void startTimer(int questionNumber) {
        showQuestion(questionNumber);
        progressIndicator.setIndicatorColor(Color.GREEN);
        new CountDownTimer(60000,1000){
            @Override
            public void onTick(long l) {
                progressIndicator.setProgress((int)l / 1000);
                progressText.setText((l / 1000) + "");
                if((l / 1000) < 15 ){

progressIndicator.setIndicatorColor(Color.RED);
                }else if((l / 1000) < 30 ){
```

20

```
progressIndicator.setIndicatorColor(Color.parseColor("#BA883E"))
;
                }
                if(!stopResponse){
                    this.cancel();
                    questionCount++;
                    loadQuestion();
                }
            }

            @Override
            public void onFinish() {
                questionCount++;
                loadQuestion();
            }
        }.start();
    }

    private void showQuestion(int questionNumber) {
        Question question =
questionArrayList.get(questionNumber);
        questionTextView.setText(question.getQuestion());
        optionA.setText(question.getOptionA());
        optionB.setText(question.getOptionB());
        optionC.setText(question.getOptionC());
        optionD.setText(question.getOptionD());
        optionA.setBackgroundResource(R.drawable.blur_white);
        optionB.setBackgroundResource(R.drawable.blur_white);
        optionC.setBackgroundResource(R.drawable.blur_white);
        optionD.setBackgroundResource(R.drawable.blur_white);
    }

    @Override
    public void onBackPressed() {
        backCount++;
        if(backCount == 2){
            super.onBackPressed();
        }
        new CountDownTimer(1500,1000){

            @Override
            public void onTick(long l) {

            }

            @Override
            public void onFinish() {
```

21

```
            backCount = 0;
        }
    }.start();
    if(backCount == 1){
        Toast.makeText(this, "Press back again to exit!",
Toast.LENGTH_SHORT).show();

    }
}


}
```

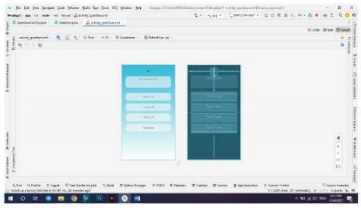
QUESTION CLASS :

```
package com.example.McqApp;

public class Question {
    String question, optionA, optionB, optionC, optionD,
correctOption, summary;

    public Question(String question, String optionA, String
optionB, String optionC, String optionD, String correctOption) {
        this.question = question;
        this.optionA = optionA;
        this.optionB = optionB;
        this.optionC = optionC;
        this.optionD = optionD;
        this.correctOption = correctOption;
    }
    public Question(String summary) {
        this.summary = summary;

    }

    public String getQuestion() {
        return question;
    }

    public String getSummary() {
        return summary;
    }

    public void setQuestion(String question) {
        this.question = question;
    }
```

22

```
    public void setSummary(String summary) {
        this.summary = summary;
    }

    public String getOptionA() {
        return optionA;
    }

    public void setOptionA(String optionA) {
        this.optionA = optionA;
    }

    public String getOptionB() {
        return optionB;
    }

    public void setOptionB(String optionB) {
        this.optionB = optionB;
    }

    public String getOptionC() {
        return optionC;
    }

    public void setOptionC(String optionC) {
        this.optionC = optionC;
    }

    public String getOptionD() {
        return optionD;
    }

    public void setOptionD(String optionD) {
        this.optionD = optionD;
    }

    public String getCorrectOption() {
        return correctOption;
    }

    public void setCorrectOption(String correctOption) {
        this.correctOption = correctOption;
    }

}
```

23

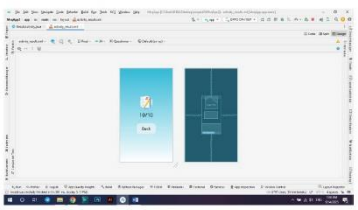


RESULT ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class ResultActivity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);

        TextView textView = findViewById(R.id.marks);
        textView.setText(getIntent().getAction());
```

24

```
        findViewById(R.id.backResult).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(ResultActivity.this,TopicActivity.class));
                ResultActivity.this.finish();
            }
        });

    }
}
```



ADMIN ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
```

25

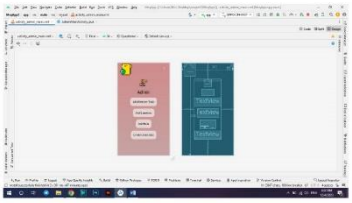
```
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.Toast;

public class AdminMainActivity extends AppCompatActivity {

    MyDbHelper dbHelper;
    ImageView img;
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_main);
        dbHelper = new MyDbHelper(this);


        findViewById(R.id.topic_more).setOnClickListener(view ->
showPopUpMenu(view));


findViewById(R.id.adminMainAddQuestions).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(AdminMainActivity.this,AddQuestionActivity.class));
            }
        });


findViewById(R.id.adminMainAddRemove).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(AdminMainActivity.this,AdminTopicActivity.class));
            }
        });

        findViewById(R.id.addnote).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

26

```
                startActivity(new
Intent(AdminMainActivity.this,AddSummaryActivity.class));
            }
        });

        findViewById(R.id.adminDemo).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences sharedPreferences =
getSharedPreferences("QUESTIONER",MODE_PRIVATE);

if(sharedPreferences.getBoolean("DemoCreated",false)){
                Toast.makeText(AdminMainActivity.this, "Demo
Quiz already created!", Toast.LENGTH_SHORT).show();
            }else {
                SharedPreferences.Editor editor =
sharedPreferences.edit();
                editor.putBoolean("DemoCreated",true);
                editor.apply();
                dbHelper.createDemoQuiz();
                Toast.makeText(AdminMainActivity.this,
"Created!", Toast.LENGTH_SHORT).show();
            }
        }
    });


        img = findViewById(R.id.show_results);
        img.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(AdminMainActivity.this,StudentMarkActivity.class));
            }
        });

    }

    private void showPopUpMenu(View view){
        PopupMenu popupMenu = new
PopupMenu(AdminMainActivity.this,view);
        MenuInflater inflater = popupMenu.getMenuInflater();

inflater.inflate(R.menu.topic_screen_menu,popupMenu.getMenu());
```

27

```
        popupMenu.show();
        popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                startActivity(new
Intent(AdminMainActivity.this,LoginActivity.class));
                AdminMainActivity.this.finish();
                return true;
            }
        });
    }
}
```



ADD/REMOVE TOPIC ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.View;
```

28

```
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;

public class AdminTopicActivity extends AppCompatActivity {

    EditText topicEditText;
    MyDbHelper dbHelper;
    RecyclerView topicsRecyclerView;
    TopicsAddRemoveAdapter topicsAddRemoveAdapter;
    ArrayList<String> topics;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_topic);

        dbHelper = new MyDbHelper(this);

        topics = new ArrayList<>();
        getAllTopics();
        topicsAddRemoveAdapter = new
TopicsAddRemoveAdapter(this, topics, new MyOnClickListener() {
            @Override
            public void onClick(int position) {

deleteTopicFromDatabase(topics.get(position),position);
            }
        });

        topicsRecyclerView =
findViewById(R.id.adminTopicRecycler);
        topicsRecyclerView.setAdapter(topicsAddRemoveAdapter);
        topicsRecyclerView.setLayoutManager(new
LinearLayoutManager(this));



        topicEditText = findViewById(R.id.adminTopicEditText);


findViewById(R.id.adminTopicAddBtn).setOnClickListener(new
View.OnClickListener() {
```

29

```
            @Override
            public void onClick(View view) {

addTopicToDatabase(topicEditText.getText().toString().trim());
            }
        });
    }

    private void getAllTopics() {
        topics.clear();
        topics = dbHelper.getAllTopics();
    }

    private void deleteTopicFromDatabase(String topic,int
position) {
        if(dbHelper.deleteTopic(topic) == -1){
            Toast.makeText(this, "Error!",
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "Topic Deleted!",
Toast.LENGTH_SHORT).show();
            topics.remove(position);
            topicsAddRemoveAdapter.notifyItemRemoved(position);
        }
    }

    private void addTopicToDatabase(String topicName) {
        long result = dbHelper.createTopic(topicName);
        if(result == -1){
            Toast.makeText(this, "Error Occurred!",
Toast.LENGTH_SHORT).show();
        }else {
            Toast.makeText(this, "Topic Added!",
Toast.LENGTH_SHORT).show();
            topicEditText.setText("");
            topics.add(topicName);

topicsAddRemoveAdapter.notifyItemInserted(topics.size());
        }
    }
}
```

TOPICS ADD REMOVE ADAPTER :

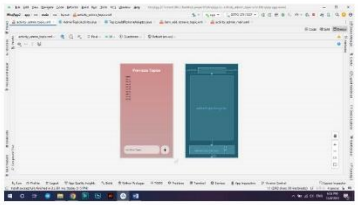
```
package com.example.McqApp;
```

30

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class TopicsAddRemoveAdapter extends
RecyclerView.Adapter<TopicsAddRemoveAdapter.MyViewHolder> {

    ArrayList<String> topics;
    Context context;
    MyOnClickListener myOnClickListener;

    public TopicsAddRemoveAdapter(Context context,
ArrayList<String> topics,MyOnClickListener myOnClickListener){
        this.topics = topics;
        this.context = context;
        this.myOnClickListener = myOnClickListener;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        return new
MyViewHolder(LayoutInflater.from(context).inflate(R.layout.item_
add_remove_topic, parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder,
int position) {

holder.topicNameTextView.setText(topics.get(holder.getAdapterPos
ition()));
        holder.trash.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

myOnClickListener.onClick(holder.getAdapterPosition());
```

31

```
            }
        });
    }

    @Override
    public int getItemCount() {
        return topics.size();
    }

    protected static class MyViewHolder extends
RecyclerView.ViewHolder{


        TextView topicNameTextView;
        ImageView trash;
        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            topicNameTextView =
itemView.findViewById(R.id.item_add_remove_topic);
            trash =
itemView.findViewById(R.id.add_remove_trash);
        }
    }
}
```



32

ADD QUESTION ACTIVITY CODE

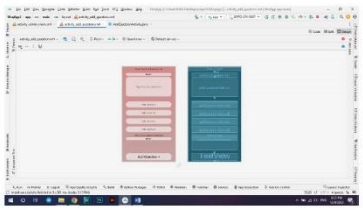
```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

public class AddQuestionActivity extends AppCompatActivity {

    MyDbHelper myDbHelper;
    EditText optionAEditText,optionBEditText, optionCEditText,
optionDEditText, questionEditText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_question);

        myDbHelper = new MyDbHelper(this);

        questionEditText =
findViewById(R.id.addQuestionEditText);
        optionAEditText = findViewById(R.id.addQuestionOptionA);
        optionBEditText = findViewById(R.id.addQuestionOptionB);
        optionCEditText = findViewById(R.id.addQuestionOptionC);
        optionDEditText = findViewById(R.id.addQuestionOptionD);

        Spinner spinner = findViewById(R.id.addQuestionSpinner);
        ArrayAdapter<String> arrayAdapterTopics = new
ArrayAdapter<>(this,
androidx.appcompat.R.layout.support_simple_spinner_dropdown_item
, myDbHelper.getAllTopics());
        spinner.setAdapter(arrayAdapterTopics);

        Spinner spinnerAnswer =
findViewById(R.id.addQuestionSpinnerAnswer);
        ArrayAdapter<String> arrayAdapterAnswer = new
ArrayAdapter<>(this,
androidx.appcompat.R.layout.support_simple_spinner_dropdown_item
, new String[]{"A", "B", "C", "D"});
        spinnerAnswer.setAdapter(arrayAdapterAnswer);
```

33

```
        spinner.setSelection(0);
        spinnerAnswer.setSelection(0);


findViewById(R.id.addQuestionButton).setOnClickListener(view ->
{
            if(arrayAdapterTopics.getCount() > 0){

addQuestionToDb(spinner.getSelectedItem().toString(),

questionEditText.getText().toString().trim(),

optionAEditText.getText().toString().trim(),

optionBEditText.getText().toString().trim(),

optionCEditText.getText().toString().trim(),

optionDEditText.getText().toString().trim(),

spinnerAnswer.getSelectedItem().toString());
            }else {
                Toast.makeText(AddQuestionActivity.this, "Please
Add Few Topics First To Proceed", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void addQuestionToDb(String topic, String question,
String optionA, String optionB, String optionC, String optionD,
String correctOption) {
        if(question.isEmpty() || optionA.isEmpty() ||
optionB.isEmpty() || optionC.isEmpty() || optionD.isEmpty()){
            Toast.makeText(this, "Fill all the fields!",
Toast.LENGTH_SHORT).show();
        }else{
            long result =
myDbHelper.addQuestion(question,optionA,optionB,optionC,optionD,
correctOption,topic);
            if(result == -1){
                Toast.makeText(this, "Error!",
Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(this, "Question Added!",
Toast.LENGTH_SHORT).show();
            }
```

34

```
        }
    }
}
```



ADD/REMOVE NOTE ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;

public class AddSummaryActivity extends AppCompatActivity {
```

35

```
    EditText noteEditText;
    MyDbHelper dbHelper;
    RecyclerView noteRecyclerView;
    NotesAddRemoveAdapter notesAddRemoveAdapter;
    ArrayList<String> notes;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_summary);


        dbHelper = new MyDbHelper(this);

        notes = new ArrayList<>();
        getAllNotes();
        notesAddRemoveAdapter = new NotesAddRemoveAdapter(this,
notes, new MyOnClickListener() {
            @Override
            public void onClick(int position) {

deleteNoteFromDatabase(notes.get(position),position);
            }
        });

        noteRecyclerView = findViewById(R.id.adminnoteRecycler);
        noteRecyclerView.setAdapter(notesAddRemoveAdapter);
        noteRecyclerView.setLayoutManager(new
LinearLayoutManager(this));

        noteEditText = findViewById(R.id.adminnoteEditText);


findViewById(R.id.adminnoteAddBtn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

addNoteToDatabase(noteEditText.getText().toString().trim());
            }
        });

    }
```

36

```
    private void getAllNotes() {
        notes.clear();
        notes = dbHelper.getAllSummaries();
    }

    private void deleteNoteFromDatabase(String note,int
position) {
        if(dbHelper.deleteSummary(note) == -1){
            Toast.makeText(this, "Error!",
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "NOTE Deleted!",
Toast.LENGTH_SHORT).show();
            notes.remove(position);
            notesAddRemoveAdapter.notifyItemRemoved(position);
        }
    }

    private void addNoteToDatabase(String notetxt) {
        long result = dbHelper.createSummary(notetxt);
        if(result == -1){
            Toast.makeText(this, "Error Occurred!",
Toast.LENGTH_SHORT).show();
        }else {
            Toast.makeText(this, "NOTE Added!",
Toast.LENGTH_SHORT).show();
            noteEditText.setText("");
            notes.add(notetxt);

notesAddRemoveAdapter.notifyItemInserted(notes.size());
        }
    }


}
```

NOTES ADD REMOVE ADAPTER :

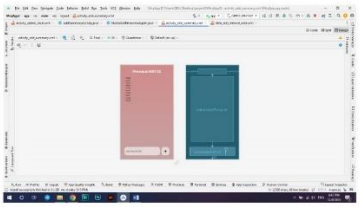
```
package com.example.McqApp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
```

37

```
import java.util.ArrayList;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class NotesAddRemoveAdapter extends
RecyclerView.Adapter<NotesAddRemoveAdapter.MyViewHolder> {

    ArrayList<String> notes;
    Context context;
    MyOnClickListener myOnClickListener;

    public NotesAddRemoveAdapter(Context context,
ArrayList<String> notes,MyOnClickListener myOnClickListener){
        this.notes = notes;
        this.context = context;
        this.myOnClickListener = myOnClickListener;
    }

    @NonNull
    @Override
    public NotesAddRemoveAdapter.MyViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new
MyViewHolder(LayoutInflater.from(context).inflate(R.layout.item_
add_remove_note, parent, false));
    }
    @Override
    public void onBindViewHolder(@NonNull
NotesAddRemoveAdapter.MyViewHolder holder, int position) {

holder.notetxtTextView.setText(notes.get(holder.getAdapterPositi
on()));
        holder.trash.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

myOnClickListener.onClick(holder.getAdapterPosition());
            }
        });
    }

    @Override
    public int getItemCount() {
        return notes.size();
```

38

```
    }

    protected static class MyViewHolder extends
RecyclerView.ViewHolder{
        TextView notetxtTextView;
        ImageView trash;
        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            notetxtTextView =
itemView.findViewById(R.id.item_add_remove_note);
            trash =
itemView.findViewById(R.id.add_remove_trash);
        }
    }
}
```



MARKS ACTIVITY CODE

```
package com.example.McqApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
```

39

```
import android.content.Intent;
import android.os.Bundle;

import java.util.ArrayList;
import java.util.List;

public class StudentMarkActivity extends AppCompatActivity {

    RecyclerView recyclerView;
    MarksAdapter marksAdapter;

    MyDbHelper dbHelper;

    ArrayList<String> markslist;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_student_mark);

        dbHelper=new MyDbHelper(this);
        recyclerView=findViewById(R.id.recycle_view);

        setRecyclerView();


    }

    private void setRecyclerView(){
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        marksAdapter= new MarksAdapter(this,getlist());
        recyclerView.setAdapter(marksAdapter);
    }
    private ArrayList<String> getlist(){
        markslist = new ArrayList<>();
        markslist = dbHelper.showResult();
        return markslist;
        }
}
```

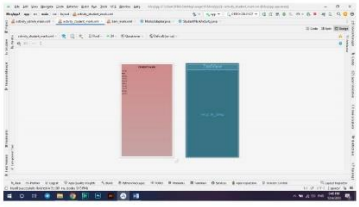
MARKS ADAPTER :

40

```
package com.example.McqApp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class MarksAdapter extends
RecyclerView.Adapter<MarksAdapter.ViewHolder> {


    Context context;
    ArrayList<String> mark_list;

    public MarksAdapter(Context context, ArrayList<String>
mark_list) {
        this.context = context;
        this.mark_list = mark_list;
    }


    @NonNull
    @Override
    public MarksAdapter.ViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View view=
LayoutInflater.from(context).inflate(R.layout.item_mark,parent,f
alse);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull
MarksAdapter.ViewHolder holder, int position) {
        if (mark_list != null && mark_list.size()>0){
            String model= mark_list.get(position);
            holder.studentmark.setText(model);
        }
        else {
            return;
```

41

```
        }

    }

    @Override
    public int getItemCount() {
        return mark_list.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView studentmark;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);

            studentmark=itemView.findViewById(R.id.studentmark);

        }
    }
}
```



DATABASE

```
package com.example.McqApp;

import android.annotation.SuppressLint;
```

42

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class MyDbHelper extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "QuestionerDB";
    public static final String TABLE_TOPIC = "TOPICS";
    public static final String TABLE_SUMMARY = "SUMMARIES";
    public static final String TABLE_USERS = "USERS";
    public static final String TABLE_USERDATA = "USERDATA";
    public static final String TABLE_QUESTIONS = "QUESTIONS";


    public MyDbHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL("CREATE TABLE USERS (username
text PRIMARY KEY, password text);");
        sqLiteDatabase.execSQL("CREATE TABLE TOPICS (topicname
text PRIMARY KEY);");
        sqLiteDatabase.execSQL("CREATE TABLE SUMMARIES
(summaryname text PRIMARY KEY);");
        sqLiteDatabase.execSQL("CREATE TABLE USERDATA (id
integer PRIMARY KEY autoincrement, username text, date text,
marks text , topic text );");
        sqLiteDatabase.execSQL("CREATE TABLE QUESTIONS (id
integer PRIMARY KEY autoincrement, question text, a text, b
text, c text, d text, answer text, topic text);");

    }
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i,
int ii) {

    }
```

43

```
    public long createUserAccount(String username,String
password){
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("username",username);
        values.put("password",password);
        long result = database.insert(TABLE_USERS,null,values);
        database.close();
        return  result;
    }


    public int loginUser(String username,String password){
        SQLiteDatabase database = this.getReadableDatabase();
        try (Cursor cursor = database.rawQuery("SELECT * FROM "
+ TABLE_USERS + " WHERE username = '" + username + "' and
password = '" + password + "'", null)) {
            if(cursor.getCount() == 1){
                cursor.close();
                return 1;
            }else {
                cursor.close();
                return -1;
            }
        }
    }

    public long createTopic(String topicName) {
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("topicname",topicName);
        long result = database.insert(TABLE_TOPIC, null,
contentValues);
        database.close();
        return result;
    }

    public long createSummary(String summaryName) {
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("summaryname",summaryName);
        long result = database.insert(TABLE_SUMMARY, null,
contentValues);
        database.close();
        return result;
    }
```

44

```
    public ArrayList<String> getAllTopics(){
        SQLiteDatabase database = this.getReadableDatabase();
        ArrayList<String> result = new ArrayList<>();
        try (Cursor cursor = database.rawQuery("SELECT * FROM
TOPICS", null)) {
            if(cursor.moveToFirst()){
                do {
                    result.add(cursor.getString(0));
                }while (cursor.moveToNext());
            }
        }
        return result;
    }

    public ArrayList<String> getAllSummaries(){
        SQLiteDatabase database = this.getReadableDatabase();
        ArrayList<String> result = new ArrayList<>();
        try (Cursor cursor = database.rawQuery("SELECT * FROM
SUMMARIES", null)) {
            if(cursor.moveToFirst()){
                do {
                    result.add(cursor.getString(0));
                }while (cursor.moveToNext());
            }
        }
        return result;
    }

    public int deleteTopic(String topic) {
        SQLiteDatabase database = this.getWritableDatabase();
        return
database.delete(TABLE_TOPIC,"topicname='"+topic+"'",null);
    }

    public int deleteSummary(String summaryName) {
        SQLiteDatabase database = this.getWritableDatabase();
        return
database.delete(TABLE_SUMMARY,"summaryname='"+summaryName+"'",n
ull);
    }

    public long addQuestion(String question, String optionA,
String optionB, String optionC, String optionD, String
correctOption, String topic) {
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues values = new ContentValues();
```

45

```
        values.put("question",question);
        values.put("a",optionA);
        values.put("b",optionB);
        values.put("c",optionC);
        values.put("d",optionD);
        values.put("answer",correctOption);
        values.put("topic",topic);
        long result =
database.insert(TABLE_QUESTIONS,null,values);
        database.close();
        return  result;
    }


    public ArrayList<Question> getQuestions(String topic){
        SQLiteDatabase database = this.getReadableDatabase();
        ArrayList<Question> result = new ArrayList<>();
        try (Cursor cursor = database.rawQuery("SELECT * FROM
QUESTIONS WHERE topic='"+ topic +"'", null)) {
            if(cursor.moveToFirst()){
                do {
                    result.add(new
Question(cursor.getString(1),cursor.getString(2),cursor.getStrin
g(3),cursor.getString(4),cursor.getString(5),cursor.getString(6)
));
                }while (cursor.moveToNext());
            }
        }
        return result;
    }

    public void addResult(String username, String date, String
marks, String topic) {
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("username",username);
        values.put("date",date);
        values.put("marks",marks);
        values.put("topic",topic);
        database.insert(TABLE_USERDATA,null,values);
        database.close();
    }

    @SuppressLint("Range")
    public ArrayList<String> showResult() {
        SQLiteDatabase database = this.getReadableDatabase();
        ArrayList<String> result = new ArrayList<>();
```



```
        try (Cursor cursor = database.rawQuery("SELECT username,
marks FROM USERDATA", null)) {
            if(cursor.moveToFirst()){
                do {

result.add(cursor.getString(cursor.getColumnIndex(cursor.getColu
mnName(0))));

result.add(cursor.getString(cursor.getColumnIndex(
cursor.getColumnName(1))));
                }while (cursor.moveToNext());
            }
        }
        return result;
    }


    public void createDemoQuiz(){
        SQLiteDatabase database = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("topicname","Algebra");
        database.insert(TABLE_TOPIC,null,values);
        values.clear();
        values.put("question","subtract: 5xy from 7xy");
        values.put("a","2xy");
        values.put("b","3y");
        values.put("c","12x");
        values.put("d","5xy");
        values.put("answer","A");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","If the algebraic term 3 x^2 y^n
is of the sixth degree,then n =");
        values.put("a","6");
        values.put("b","5");
        values.put("c","4");
        values.put("d","3");
        values.put("answer","C");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","reduce to the simples from:
6x+7y+6x-3y");
        values.put("a","10x+3y");
        values.put("b","9x-3y");
        values.put("c","10x-4y");
```



```
        values.put("d","10x+4y");
        values.put("answer","D");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","-7y is less than -10y by");
        values.put("a","-17y");
        values.put("b","-3y");
        values.put("c","3y");
        values.put("d","17y");
        values.put("answer","B");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","the coefficient of the algebraic
term (5^3)x is");
        values.put("a","3");
        values.put("b","5");
        values.put("c","25");
        values.put("d","125");
        values.put("answer","D");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","the algebraic term 5x(y^2) is of
the ..... degree. ");
        values.put("a","first");
        values.put("b","second");
        values.put("c","third");
        values.put("d","fifth");
        values.put("answer","C");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","If (x-1) is one the factors of
(x^2+ 5x- 6) ,then find the other factor");
        values.put("a","x+7");
        values.put("b","x-1");
        values.put("c","x+6");
        values.put("d","x+1");
        values.put("answer","C");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","If x+y=4, x^2+xy=24,then x=");
        values.put("a","4");
        values.put("b","6");
```



```
        values.put("c","8");
        values.put("d","24");
        values.put("answer","B");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","Divide: x^2+ x- 12 by x+4 where x
not equal -4");
        values.put("a","x-3");
        values.put("b","x+3");
        values.put("c","x-2");
        values.put("d","x+2");
        values.put("answer","A");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","Factorize by identifying the
highest common factor: 5a+15b");
        values.put("a","5(a+3b)");
        values.put("b","15(a+b)");
        values.put("c","3(a+5b)");
        values.put("d","5(a-3b)");
        values.put("answer","A");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","If 3x+4y=7, then 9x+12y=");
        values.put("a","25");
        values.put("b","7");
        values.put("c","21");
        values.put("d","10");
        values.put("answer","C");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);
        values.clear();
        values.put("question","x^2 y - x y^2 -(x-y)");
        values.put("a","x");
        values.put("b","y");
        values.put("c","2xy");
        values.put("d","xy");
        values.put("answer","D");
        values.put("topic","Algebra");
        database.insert(TABLE_QUESTIONS,null,values);


    }
}
```

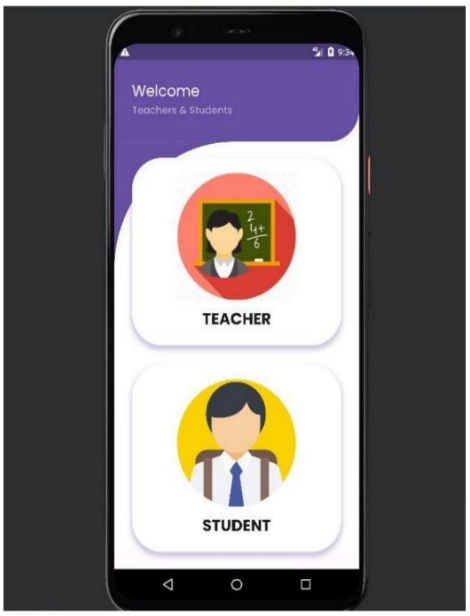
Appendix B

Appendix

Table of Contents

1

Home Page of Application




```
import 'package:flutter/material.dart';
import 'StudentPage/student_login.dart';
```

2

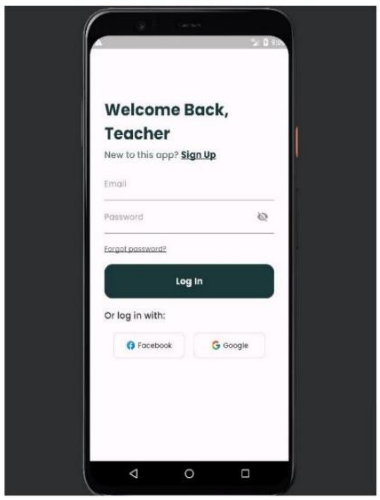
```
import 'TeacherPage/teacher_login.dart';
// Firebase
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(fontFamily: 'Poppins'),
      home: const MyHomePage(),
    );
  }
}

// Home Page
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: ListView(
        padding: EdgeInsets.zero,
        children: [
          Container(
            decoration: BoxDecoration(
              color: Theme.of(context).primaryColor,
              borderRadius: const BorderRadius.only( bottomRight: Radius.circular(50), ),
            ),
            child: Column(
              children: [
                const SizedBox(height: 50),
                ListTile(
                  contentPadding: const EdgeInsets.symmetric(horizontal: 30),
                  title: Text('Welcome', style: Theme.of(context).textTheme.headlineSmall?.copyWith(
color: Colors.white )),
                  subtitle: Text('Teachers & Students', style:
Theme.of(context).textTheme.titleMedium?.copyWith( color: Colors.white54 )),
                ),
                const SizedBox(height: 30)
              ],
            ), ),
          Container(
            color: Theme.of(context).primaryColor,
            child: Container(
              padding: const EdgeInsets.symmetric(horizontal: 30),
              decoration: const BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.only( topLeft: Radius.circular(200))
              ),
              child: GridView.count(
                shrinkWrap: true,
                physics: const NeverScrollableScrollPhysics(), crossAxisCount: 1, crossAxisSpacing: 40,
mainAxisSpacing: 30,
                childAspectRatio: 1.2,
                children: [ itemDashboard('Teacher', 'images/teacher_icon.png', Colors.deepOrange,
const LogInTeacher()),
                            itemDashboard('Student', 'images/student_icon.png', Colors.brown, const
LogInStudent()),],
```

3

```
            ),
          ),
          const SizedBox(height: 20)
        ],
      ),
    );
  }

  itemDashboard(String title, String imagePath, Color background, Widget page) => GestureDetector(
    onTap: () { // go to next page
      Navigator.push( context, MaterialPageRoute(builder: (context) => page),
      );
    },
    child: Container(
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.circular(10),
        boxShadow: [
          BoxShadow(
            offset: const Offset(0, 5),
            color: Theme.of(context).primaryColor.withOpacity(0.3),
            spreadRadius: 2,
            blurRadius: 5,
          )
        ],
      ),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Image.asset( imagePath, height: 200, ), // add image to teacher button or student button
          const SizedBox(height: 10),
          Text(
            title.toUpperCase(),
            style: Theme.of(context).textTheme.titleMedium?.copyWith(
              fontSize: 25, // Adjust the font size as needed
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    ),
  );
}
```

4

Teacher Login Form




```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:teaching_project/TeacherPage/teacher_screen.dart';
import 'package:teaching_project/TeacherPage/teacher_reset_password.dart';
import 'package:teaching_project/TeacherPage/teacher_signup.dart';
import 'package:teaching_project/theme.dart';
import 'package:teaching_project/TeacherPage/teacher_login_option.dart';
import 'package:teaching_project/widgets/primary_button.dart';
```

5

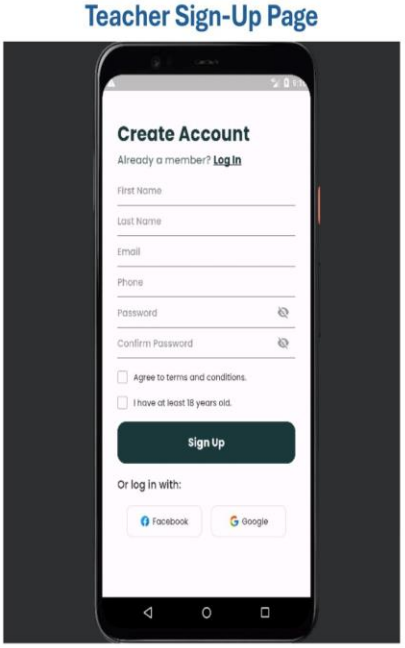
```
class LogInTeacher extends StatefulWidget {
  const LogInTeacher({super.key});
  @override
  _LogInScreenState createState() => _LogInScreenState();
}

class _LogInScreenState extends State<LogInTeacher> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  bool _isChecked = true;

  signInWithEmailAndPassword() async {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
          email: _emailController.text,
          password: _passwordController.text,
      );
      // Clear data from login page
      _emailController.clear();
      _passwordController.clear();
      // go to Teacher Home Page
      Navigator.push( context, MaterialPageRoute(builder: (context) => TeacherScreen()),
      );

    } on FirebaseAuthException catch (e) {
      if(e.code == 'invalid-email'){
        return ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Invalid
Email."),));
      } else if (e.code == "user-not-found") {
        ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("No user found for
that email."),));
      } else if (e.code == "wrong-password") {
        ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Wrong password
provided for that user."),));
      } else {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("An error occurred:
${e.message}")),);
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: kDefaultPadding,
        child: SingleChildScrollView(
          child: Form(
            key: _formKey,
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                const SizedBox(height: 125,),
                Text('Welcome Back, Teacher', style: titleText,),
                const SizedBox(height: 5,),
                Row(
                  children: [
                    Text('New to this app?', style: subTitle,),
                    const SizedBox(width: 5,),
                    GestureDetector(
                      onTap: () { // Go to Sign Up Page -> For Teacher
                        Navigator.push(context, MaterialPageRoute(builder: (context) => const
SignUpScreen(),),);

                        // Clear data from login page
                        _emailController.clear();
                        _passwordController.clear();
                      },
                      child: Text('Sign Up',
                          style: textButton.copyWith(decoration: TextDecoration.underline,
decorationThickness: 1,),),),
                  ],
                ),
```

6

```
            const SizedBox(height: 10,),

            // Login Form //
            TextFormField( // text of email
              controller: _emailController,
              validator: (text){
                if (text == null || text.isEmpty) {
                  return 'Email is empty';
                } else if (!RegExp(r"^dr\..*\@gmail\.com$").hasMatch(text)) {
                  return 'Email should start with "dr." and end with "@gmail.com"';
                } else {
                  return null;
                }
              },
              obscureText: false ? _isObscure : false,
              decoration: const InputDecoration(labelText: "Email", labelStyle: TextStyle(color:
kTextFieldColor),),
                focusedBorder: UnderlineInputBorder(borderSide: BorderSide(color: kPrimaryColor),),
              ),
            ),
            TextFormField( // text of password
              controller: _passwordController,
              validator: (text){
                if(text == null || text.isEmpty){
                  return 'Password is empty';
                } else if (text.length < 6) {
                  return 'Password should not less then 6 characters';
                } else {
                  return null;
                }
              },
              obscureText: true ? _isObscure : false,
              decoration: InputDecoration(labelText: "Password", labelStyle: const TextStyle(color:
kTextFieldColor),),
                focusedBorder: const UnderlineInputBorder(borderSide: BorderSide(color:
kPrimaryColor),),
                 suffixIcon: true
                    ? IconButton(
                    onPressed: () {
                      setState(() {
                        _isObscure = !_isObscure;
                      });
                    },
                    icon: _isObscure
                        ? const Icon(
                      Icons.visibility_off,
                      color: kTextFieldColor,
                    )
                        : const Icon(
                      Icons.visibility,
                      color: kPrimaryColor,
                    ),
                  )
                    : null,
              ),
            ),
            const SizedBox(height: 20,),

            GestureDetector(
                onTap: () {Navigator.push(context, MaterialPageRoute(builder: (context) => const
ResetPasswordScreen()));},
                child: const Text('Forgot password?',
                  style: TextStyle(color: kSambalColor, fontSize: 14, decoration:
TextDecoration.underline, decorationThickness: 1,),
                ),
            ),
            const SizedBox(height: 20,),


            // Button Function //
            PrimaryButton(buttonText: 'Log In',
              onPressed: () {
                if(_formKey.currentState!.validate()){  // check if email and password field is
empty  or not
                  signInWithEmailAndPassword();
```



```
                },
              ),
              const SizedBox(height: 20,),

              Text('Or log in with:', style: subTitle.copyWith(color: kBlackColor),),
              const SizedBox(height: 20,),

              const LoginOption(),
            ],
          ),
        ),
      ),
    );
  }


}
```

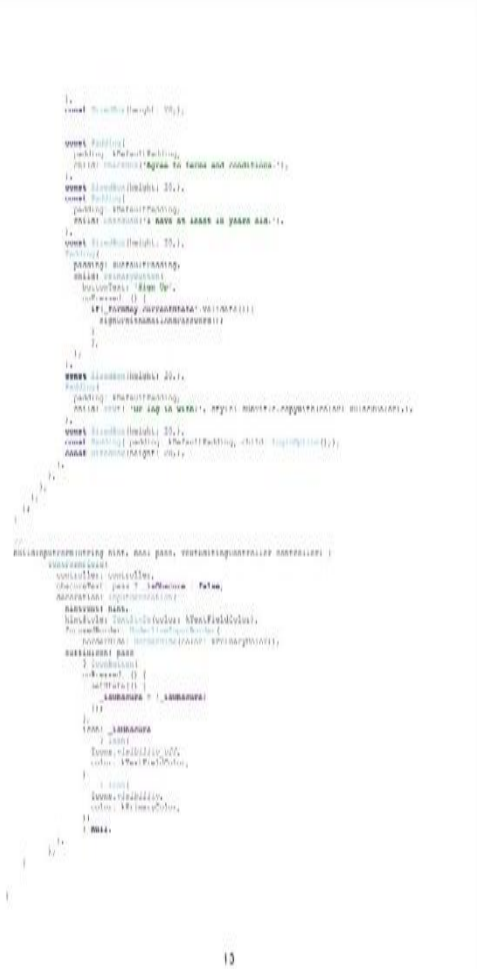

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:teaching_project/theme.dart';
import 'package:teaching_project/widgets/checkbox.dart';
import 'package:teaching_project/TeacherPage/teacher_login_option.dart';
import 'package:teaching_project/widgets/primary_button.dart';

class SignUpScreen extends StatefulWidget {
  const SignUpScreen({super.key});

  @override
  SignUpScreenState createState() => SignUpScreenState();
}

class SignUpScreenState extends State<SignUpScreen> {
  bool _isObscure = true;
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _firstName = TextEditingController();
  final TextEditingController _lastName = TextEditingController();
  final TextEditingController _phone = TextEditingController();
  final TextEditingController _email = TextEditingController();
  final TextEditingController _password = TextEditingController();
  final TextEditingController _confirmPassword = TextEditingController();

  signUPWithEmailAndPassword() async {
    try {
      await FirebaseAuth.instance.createUserWithEmailAndPassword(
        email: _email.text,
        password: _password.text,
      );

      // Create a new user document in Firestore
      FirebaseFirestore.instance.collection('user').doc(_email.text).set({
        'firstName': _firstName.text,
        'lastName': _lastName.text,
        'phone': _phone.text,
        'password': _password.text,
      });

      // when create account -> go to login page
      Navigator.pop(context);

      print("success");
      return ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text("Account is created successfully"),
          )
      );

    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        return ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("No user found for that email."),
            )
        );
      } else if (e.code == 'wrong-password') {
        return ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("Wrong password provided for that user."),
            )
        );
      }
    }
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const SizedBox(height: 70,),
              Padding(
```



```
                padding: kDefaultPadding,
                child: Text('Create Account', style: titleText,),
              ),
              const SizedBox(height: 5,),
              Padding(
                padding: kDefaultPadding,
                child: Row(
                  children: [
                    Text('Already a member?', style: subTitle,),
                    const SizedBox(width: 5,),
                    GestureDetector(
                      onTap: () {Navigator.pop(context); }, // to return to login
                      child: Text('Log In',
                        style: textButton.copyWith( decoration: TextDecoration.un
decorationThickness: 1,),
                      ),
                    )
                  ],
                ),
              ),
              const SizedBox(height: 10,),

              // Form of Sign Up ///
              Padding(
                padding: kDefaultPadding,
                child: Column(
                  children: [
                    // first name field
                    TextFormField( controller: _firstName,
                      validator: (text){
                        if(text == null || text.isEmpty){
                          return 'first name is empty';
                        } else {
                          return null;
                        }
                      },
                      obscureText: false ? _isObscure : false,
                      decoration: const InputDecoration( hintText: "First Name",
                      hintStyle: TextStyle(color: kTextFieldColor),
                      focusedBorder: UnderlineInputBorder(
                          borderSide: BorderSide(color: kPrimaryColor)),
                      ),
                    ),

                    // last name field
                    TextFormField( controller: _lastName,
                      validator: (text){
                        if(text == null || text.isEmpty){
                          return 'last name is empty';
                        } else {
                          return null;
                        }
                      },
                      obscureText: false ? _isObscure : false,
                      decoration: const InputDecoration( hintText: "Last Name",
                        hintStyle: TextStyle(color: kTextFieldColor),
                        focusedBorder: UnderlineInputBorder(
                            borderSide: BorderSide(color: kPrimaryColor)),
                      ),
                    ),

                    // 3 email field
                    TextFormField(
                      controller: _email,
                      validator: (text){
                        if (text == null || text.isEmpty) {
                          return 'email is empty';
                        } else if (!RegExp(r"^dr\..*\@gmail\.com$").hasMatch(text
                          return 'Email should start with "dr." and end with "@gm
                        } else {
                          return null;
                        }
                      },
                      obscureText: false ? _isObscure : false,
                      decoration: const InputDecoration(
                        hintText: "Email",
```



```
                        hintStyle: TextStyle(color: kTextFieldColor),
                        focusedBorder: UnderlineInputBorder(
                            borderSide: BorderSide(color: kPrimaryColor)),
                      ),
                    ),

                    // 4 phone field
                    TextFormField(
                      controller: _phone,
                      validator: (text){
                        if (text == null || text.isEmpty) {
                          return 'phone is empty';
                        } else if (text.length != 11) {
                          return 'phone should be 11 number';
                        } else {
                          return null;
                        }
                      },
                    ),

                    // 5 password field
                    TextFormField(
                      controller: _password,
                      validator: (text){
                        if (text == null || text.isEmpty) {
                          return 'password is empty';
                        } else if (text.length < 6) {
                          return 'password should be less than 6 character';
                        } else {
                          return null;
                        }
                      },
                    ),

                    // 6 confirm password field
                    TextFormField(
                      controller: _confirmPassword,
                      validator: (text){
                        if (text == null || text.isEmpty) {
                          return 'password is empty';
                        } else if (text.compareTo( password.text) != 0) {
                          return 'password does not match';
                        } else if (text.length < 6) {
                          return 'password should be less than 6 characters';
                        } else {
                          return null;
                        }
                      },
                      obscureText: true ? _isObscure : false,
                      decoration: InputDecoration(
                        hintText: "Confirm Password",
                        hintStyle: const TextStyle(color: kTextFieldColor),
                        focusedBorder: const UnderlineInputBorder(
                            borderSide: BorderSide(color: kPrimaryColor)),
                        suffixIcon: true
                            ? IconButton(
                            onPressed: () {
                              setState(() {
                                _isObscure = !_isObscure;
                              });
                            },
                            icon: _isObscure
                                ? const Icon(
                              Icons.visibility_off,
                              color: kTextFieldColor,
                            )
                                : const Icon(
                              Icons.visibility,
                              color: kPrimaryColor,
                            ))
                            : null,
                      ),
                    ),
                  ],
                )
```

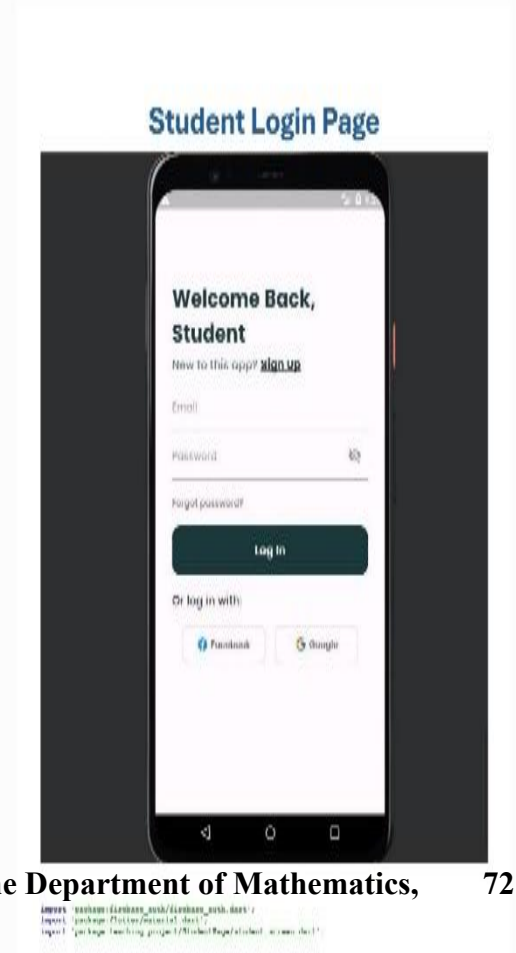


13

14

15

16

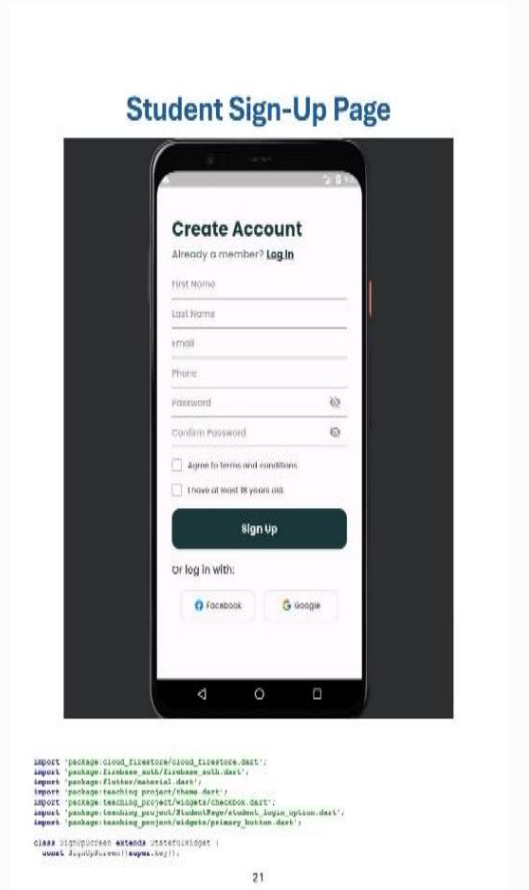


17

18

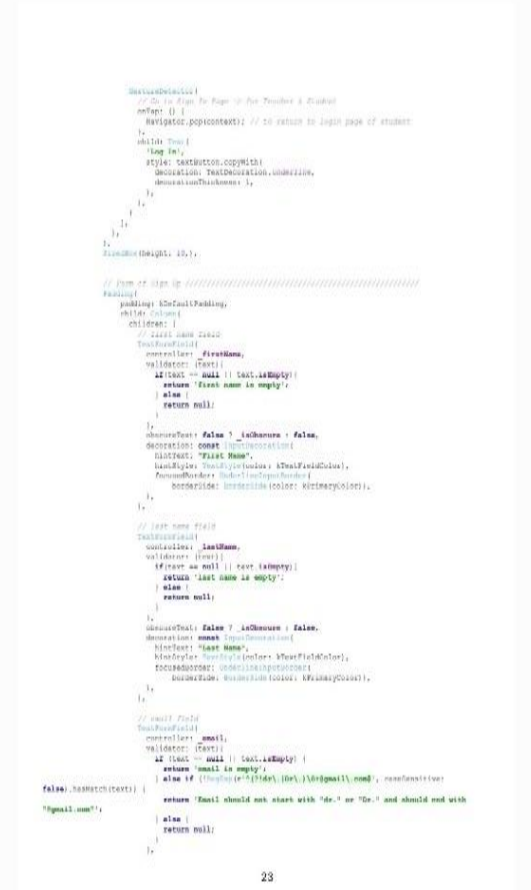
19

20

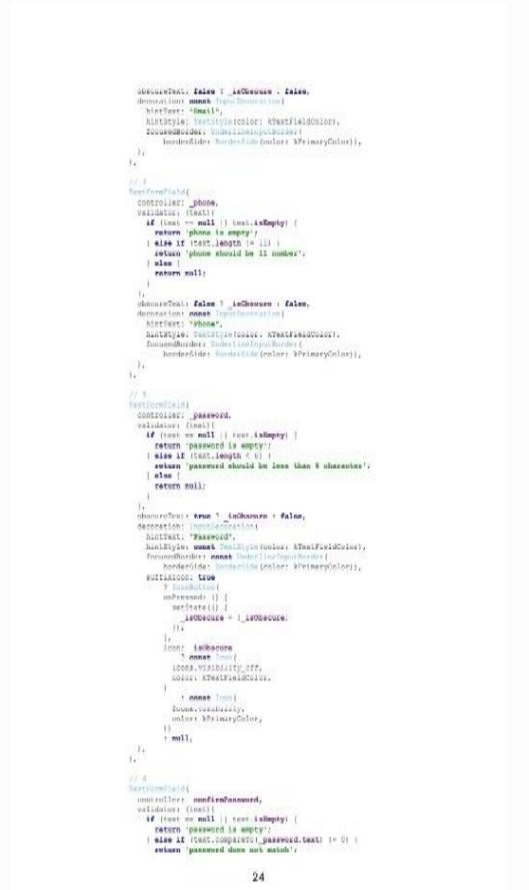


21

22



23



24

Student Home Page


```
ge:cloud_firestore/cloud_firestore.dart';
ge:firebase_auth/firebase_auth.dart';
ge:firebase_database/firebase_database.dart';
```



```
'package:flutter/material.dart';
'package:teaching_project/StudentPage/student_login.dart';
'package:teaching_project/widgets/primary_button.dart';
'package:url_launcher/url_launcher.dart';


StudentScreen extends StatefulWidget {
t StudentScreen({super.key});

rride
dentScreenState createState() => _StudentScreenState();


_StudentScreenState extends State<StudentScreen> {

l user = FirebaseAuth.instance.currentUser;
url = 'https://docs.google.com/forms/u/3/';

nchURL() async {
y {
// Fetch data from Firebase
// if the app can not open the link of exam , it will open empty google form
final examDoc = await FirebaseFirestore.instance.collection('URL').doc('exam').get();

if (examDoc.exists) {
  final examData = examDoc.data()!;
  final examUrl = examData['examURL'];
  print('------>> $examUrl');
  if (await canLaunchUrl(Uri.parse(examUrl))) {
    await launchUrl(Uri.parse(examUrl));
  } else {
    await launchUrl(Uri.parse(url));
    throw 'Could not launch url => $examUrl';
  }
} else {
  throw 'Exam document not found';
}
catch (e) {
print('Error: $e');


rride
et build(BuildContext context) {
turn Scaffold(
ppBar: AppBar(
  centerTitle: true,
  title: const Text("Student Home Page"),
  actions: [
    IconButton(
      onPressed: () async {
        await FirebaseAuth.instance.signOut();
        Navigator.pop(context);   // to return to login page of student
      },
      icon: const Icon(Icons.login),
    )
  ],
),
body: Center(
  child: PrimaryButton( buttonText: 'Open Exam', onPressed: _launchURL, ),
),
```

Appendix C

Student

Q1. I have encountered a technical issue while using electronic quizzes

Q2. I find electronic quizzes more engaging and interactive than traditional quizzes

Q3. I noticed some improvement in my retention and understanding of the subject matter when using electronic quizzes.

Q4. I would prefer taking electronic quizzes over paper quizzes.

Q5. It's easy to understand the mathematical concepts presented in electronic quizzes.

Q6. It's easy to understand the mathematical concepts presented in paper quizzes.

Q7. I have an ability to manage time during an electronic quiz more than paper quiz.

Q8. I have an ability to manage time during a paper quiz.

Q9. I feel focused when taking an electronic quiz.

Q10. I feel focused when taking a paper quiz.

Q11. Mathematics electronic quizzes enhance your problem-solving skills.

Teacher

Q1. Preparation for electronic quizzes is more effectively than the paper quizzes.

Q2. Preparation for paper quizzes is more effectively than electronic quizzes.

Q3. Electronic quizzes are a valuable tool for assessing student learning.

Q4. Electronic quizzes have improved the efficiency of my assessment practices.

Q5. I find electronic quizzes more engaging for students than traditional paper quizzes.

Q6. My students provide positive feedback about electronic quizzes.

Q7. The platform allows me to easily create and modify quizzes to fit my specific learning objective.

Q8. My students can access and complete electronic quizzes easily from any device.

Q9. I'm able to provide immediate feedback to students on their quiz performance.

Q10. Calculation of marks are better in online exams than written exams.

Q11. It is fair if students take the final exams online.

Parents

Q1. My child uses electronic quizzes for independent learning.

Q2. My child has easy access to a variety of electronic quizzes at home.

Q3. Electronic quizzes are an effective way for my child to learn new information

Q4. Electronic quizzes help my child to retain information better than traditional paper quizzes.

Q5. Electronic quizzes provide immediate feedback that helps my child learn from their mistakes.

Q6. Electronic quizzes motivate my child to learn and improve.

Q7. I worry that my child may become overly reliant on electronic quizzes for learning.

Q8. I find it challenging to monitor my child's use of electronic quizzes.

Q9. I interested in receiving feedback from the teacher about my child's performance on electronic quizzes.

Q10. I believe that electronic quizzes can be a helpful tool for educational purposes.

Q11. My child is always worried about exams.

Q12. Online exams will be less stressful to my child.

Q13. My child scorers better on online exams than paper exams.