



## QuizUp - A Quiz Application

### Project Description

This application will provide convenience in the MCQ mode of examinations and screening tests. This application will eliminate paperwork and will be able to effectively store all the information in the database. The purpose of this application is to save student data, measure the marks obtained by each student, and point out their mistakes.

### Author

[Atanu Paul](#)

### Collaborator(s)

[Ayush Kumar Shaw](#), [Mridula Reddy](#)

### Project Language(s)

Java

### Difficulty

Intermediate

### Duration

60 hours

### Prerequisite(s)

Java Basics, Android Basics

### Skills to be learned

Firebase Data Handling

### Overview

### Objective

Creating an android application through which students can test for their exam preparation.



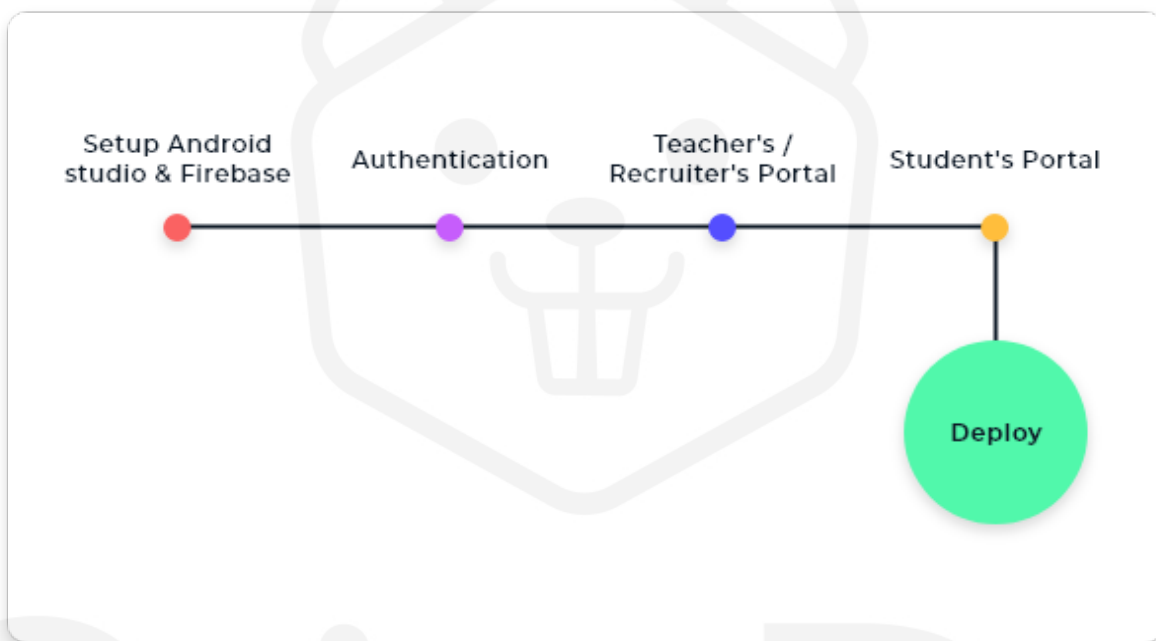
## Project Context

The quiz format for an examination has been a standard since a long time and still persists to be. The reason for the same will be the efficiency and feasibility it carries with it.

Our application is a simple real-world android application where students can take their test in a quiz format. The application is developed using Java and integrated with Firebase.

## Project Stages

The project consists of the following stages:



## High-Level Approach

- The first task will be to set up **Android Studio & Firebase**.
- Once we have everything in place, we can start off with building the authentication on login functionality.
- We have to manage two things very well in our project, one is the **Teacher/Recruiter** (who will create questions for the quiz) and another is the **Student** (who will participate in the quiz).
- We need support for database handling and authentication. So we'll use here Firebase. Basically, the database will be used to store the login information for the users, but the resource can be used for storing quizzes and student's progress history.



- The core implementation of our application constitutes successful implementation of the above-given requirements including deployment.

## Primary goals

- Create a login page for **Teacher/Recruiter** and another for **Student**.
- Create a home page for **Teacher/Recruiter** through which they can upload questions for a new quiz and see the marks of students who have attended their previous quizzes.
- Add functionalities like add questions, delete questions for **Teachers/Recruiters**.
- Create a home page for students where they can view available quizzes so that they have the opportunity to participate in those quizzes.
- Add functionalities like see progress history for students.

## Task 1

### Environment & Firebase setup

Before the start of any development procedure, we need to set up the environment according to our application needs. Then connect our Android Studio with Firebase.

### Requirements

- Install Android Studio on your machine.
- Install and set up JDK.
- Create a new Android project.
- Like any typical application, the source code of java should be in a java folder and the source code of XML should be in a res folder.
- Connect with firebase.

### References

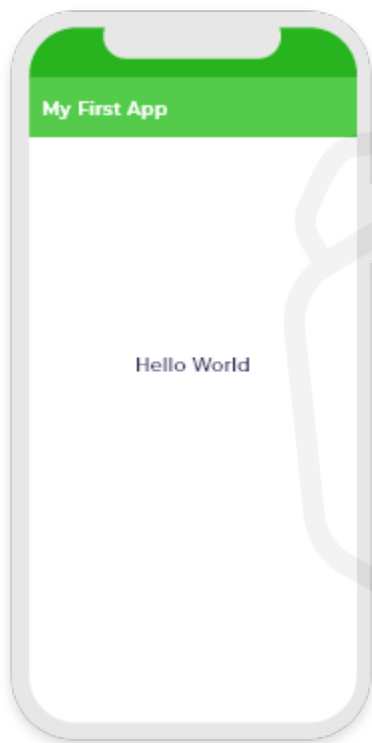
- [Download Android Studio](#)
- [Download Jdk & JRE](#)
- [Create a new project](#)
- [Create a virtual device on your machine](#)
- [A beginners guide to Gradle](#)
- [Run your application](#)
- [Connect your App to Firebase](#)



## Expected Outcome

The main objective of this milestone is to make sure that you have the required development environment in place.

On completion of the above requirements, run the application using virtual device and the end result should be as shown in the picture below.



## Task 2

### Setting up Firebase and Authentication

Firebase is a service provided by Google for configuring the backend of any application with all the general necessities like database preparation, authentication using various methods, etc. In this milestone, we'll be preparing our database and setting up authentication using email and passwords.

[**Note:** Use the references provided to implement the following requirements.]

### Requirements

- Setup sign-in method using Email/Password.



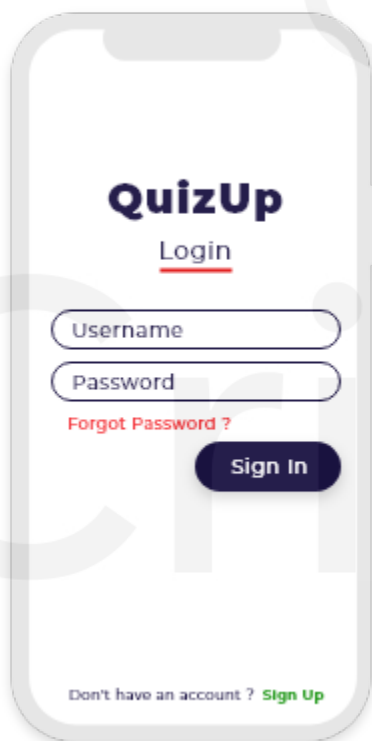
- Declare the dependency for the Firebase Authentication Android library in your module ( app-level ) Gradle file ( usually app/build.gradle ).

```
dependencies { // Import for the Firebase platform implementation
platform('com.google.firebase:firebase-bom:26.3.0')

// Declare the dependency for the Firebase Authentication library
// When using the BoM, you don't specify versions in Firebase library
dependencies
implementation 'com.google.firebase:firebase-auth'

}
```

- To use an authentication provider, you need to enable it in the Firebase console. Go to the Sign-in Method page in the Firebase Authentication section to enable Email/Password sign-in and any other identity providers you want for your app.
- Android has Material Components, which helps in building our frontend.
- Create a new activity named Login. Style the activity so that it looks similar to the one shown below.



[Note: You can design as you like.]



## References

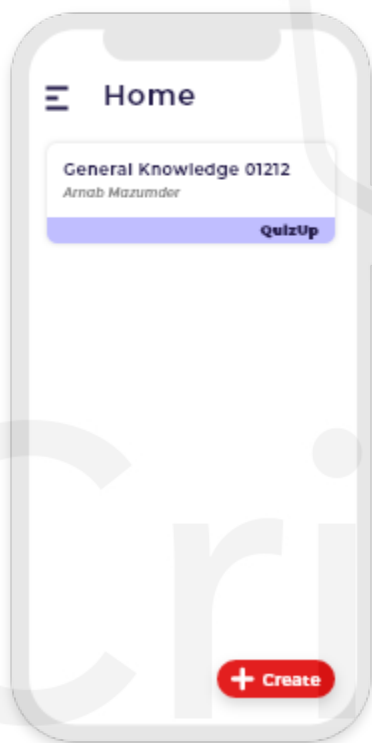
- [Material Components for Android](#)
- [Get Started with Firebase Authentication on Android](#)

## Task 3

### Creating home activity

Now we will create an activity where there will be two options **Create a new quiz** and **quiz history**.

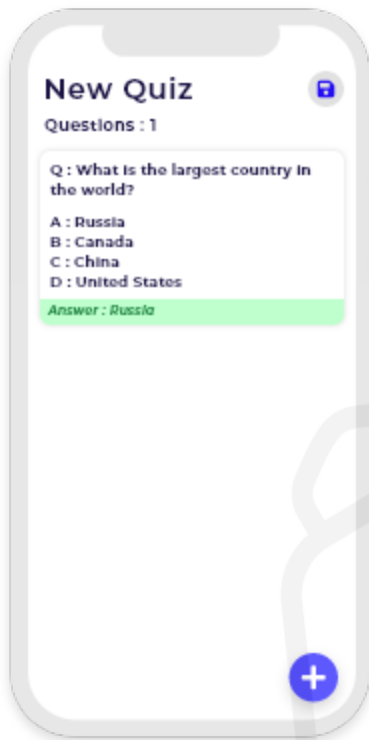
- Through the quiz history section, teachers/recruiters will be able to see the quizzes created before by them and the score of the students who participated in those quizzes.
- Style the activity so that it looks similar to the one shown below.



[**Note:** You can design as you like.]

### Creating quiz activity

Now we will create one more activity through which all the questions will be managed i.e add or delete.



[**Note:** You can design as you like.]

## Requirements

- Create a new component called App-Bar in-home activity.
- Add the necessary code in the xml and java files so that all the work is done accurately as mentioned earlier.

## Tip

You can use Listview or Recycleview to show questions as mentioned earlier. Using Listview or Recycleview, you can show the questions and their options with answers.

## References

- [Material Components for Android](#)
- [Read and Write Data into Firebase Database](#)



## Task 4

### Creating another home activity

Now we will create an activity where there will be another two options: **Available Quizzes** and **Progress history**.

- The available quiz option allows students to participate in available quizzes.
- In the Progress history section, students will be able to see the marks of the quiz they have attended before.
- Students will be able to search for available quizzes by using search features.
- Style the activity so that it looks similar to the one shown below.



[Note: You can design as you like.]

### Creating quiz activity

After choosing any one of the available quizzes in Home Activity, students will be able to see some information about the quiz such as the name of the quiz, what should be informed about taking the quiz, how long can this quiz be given, etc. After viewing all of these and clicking the start button, a student will be able to take that quiz.

- The timer will start as soon as the quiz starts so that the students can see the back counting on the timer.



- After choosing any one of the options of a question, the next question will appear on the screen and thus the test will continue till all the questions are answered.

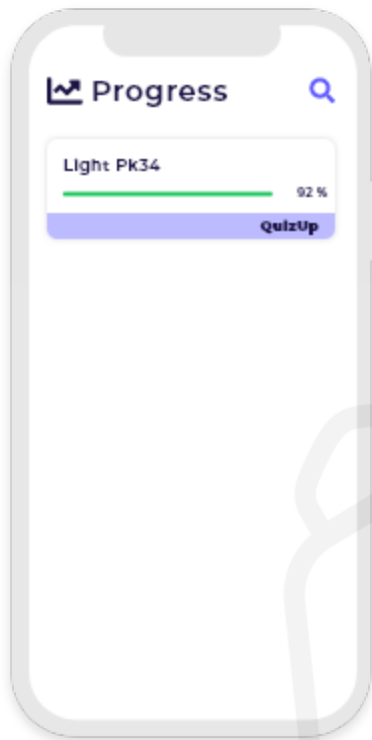
[**Note:** If a student is unable to answer all the questions before the time countdown ends, the test will automatically close and the result will be shown as the number of answers.]



[**Note:** You can design as you like.]

### Creating progress activity

Now we will create a progress activity through which students can see the results of their previous quiz here.



[Note: You can design as you like.]

## Requirements

- Create a new component called App-Bar in home , exam & history activity.
- Add the necessary code in the xml and java file so that all the work is done accurately as mentioned earlier.
- Create a timer function for back counting.

```
new CountdownTimer(30000, 1000) {  
    public void onTick(long millisUntilFinished) {  
        mTextField.setText("seconds remaining: " + millisUntilFinished /  
1000);  
    }  
  
    public void onFinish() {  
        mTextField.setText("Time Up!");  
    }  
}.start();
```



## References

- [Material Components for Android](#)
- [Read and Write Data into Firebase Database](#)
- [CountDown Timer](#)

## Tip

- You can use Listview or Recycleview to show quizzes and progress history as mentioned earlier.

## Expected Outcome

By the end of this milestone, you should have a functional quiz app, applicable for teachers/recruiters as well as students.

