



Database Management System

Mini Project Report

January - June 2023

Quest
A Quiz Application

Team Members:

S.Deepanraj– 2021103732

A.Divyesh Prasad – 2021103523

S.Kanishka - 2021103536

Abstract:

The Quest application is a web-based application that allows users to participate in quizzes and assess their knowledge on various topics. It provides a user-friendly interface for taking quizzes, tracking scores, and managing quiz. The purpose of the Quest application is to provide an engaging and interactive platform for users to test their knowledge and learn through quizzes. It can be used in educational settings, training programs, or for recreational purposes.

The app allows users to register, log in, and select quizzes to take. It supports different question formats such as multiple-choice and true/false, and provides immediate feedback on the user's performance. The app includes features such as timed quizzes, score tracking, leaderboard comparisons, and an admin panel for quiz management. The Quiz App offers an engaging and educational experience for users to test their knowledge, learn new information, and compete with others. The app is built using modern web technologies such as Flutter for the frontend and Firebase for the backend and database.

Introduction:

In today's modern world, where technology is pervasive and education is evolving, the Quiz App plays a crucial role in facilitating learning and knowledge acquisition. It leverages the power of technology to create interactive, engaging, and personalized learning experiences. The Quiz App has become increasingly relevant in today's fast-paced and digitally-driven society. It aligns with the changing dynamics of education and the needs of learners.

The Quest App breaks the barriers of time and location by providing access to quizzes anytime, anywhere. With the app accessible on various devices such as smartphones, tablets, and computers, learners can engage in quizzes at their convenience. This accessibility accommodates the busy schedules of individuals and promotes continuous learning. Every learner has unique needs and preferences. The Quest App allows for personalized learning experiences by offering a wide range of quizzes on diverse topics. Users can choose quizzes that align with their interests, educational goals, or areas of focus. This personalization fosters engagement, as learners are more likely to invest time and effort in topics that resonate with them.

By answering quiz questions, learners actively participate in the learning process, retrieve knowledge from memory, and engage in critical thinking. This interactivity enhances understanding, retention, and application of knowledge. In this Quiz App, learners receive instant feedback on their quiz performance. This immediate feedback helps identify strengths and weaknesses, correct misconceptions, and guide further learning. Additionally, progress tracking features allow learners to monitor their growth, set goals, and track their improvement over time. It empowers learners to engage actively, receive immediate feedback, track their progress, and enjoy the process of acquiring knowledge. As technology continues to shape the educational

landscape, the Quiz App stands as a valuable tool for lifelong learning, fostering intellectual growth, and promoting a culture of continuous education in today's dynamic and ever-changing world.

Environmental Support:

An application is developed with the help of both Front-end languages and Back-end languages where the designing of user interface and the functionality of the app are done with these languages. The Quest app is developed with the help of Flutter and Firebase.

Flutter is an open-source UI software development kit (SDK) developed by Google. It is used for building high-quality, cross-platform applications for mobile, web, and desktop platforms from a single codebase. The choice of Flutter as the development framework and integrating it with a robust database solution like Firebase provides extensive environmental support for our Quiz App project. Flutter's cross-platform capabilities enable us to build the app once and deploy it seamlessly on various platforms, including iOS, Android, web, and desktop. This cross-platform support ensures maximum reach and accessibility for our app, allowing users to engage with quizzes on their preferred devices. Additionally, Flutter's native-like performance ensures a smooth and responsive user experience, with animations and interactions that closely resemble those of native applications. This ensures that our Quiz App feels fast, intuitive, and seamless, regardless of the platform it is running on.

Firebase is a comprehensive mobile and web development platform provided by Google. It offers a wide range of backend services and tools that help developers build, deploy, and manage applications efficiently. Integrating the database solution brings significant benefits to our Quiz App. Firebase offers real-time data synchronization, enabling users to seamlessly access their quiz progress, scores, and other data across multiple devices. The Firebase backend services, including the real-time database and cloud storage, provide scalability and reliability, ensuring that our app can handle a large number of concurrent users and accommodate traffic spikes without compromising performance or availability.

Furthermore, Firebase's authentication services enable secure user registration, login, and data access, ensuring that user information is protected. The Firebase Analytics feature allowed us to gain valuable insights into user behavior, app usage, and performance metrics, empowering us to make data-driven decisions to enhance the app's functionality and user experience.

In summary, the combination of Flutter and Firebase as the environmental support for our Quiz App project ensures cross-platform compatibility, native-like performance, real-time data synchronization, scalability, and security. This powerful combination enabled us to create a high-quality, user-friendly Quiz App that can be

accessed and enjoyed by users on various devices, while providing a seamless and engaging quiz-taking experience.

System Requirements:

1. Operating System:

- Android: Requires Android 6.0 (Marshmallow) or later.
- iOS: Requires iOS 11.0 or later.
- Windows: Requires Windows 10 (version 1809) or later.
- macOS: Requires macOS 10.13 (High Sierra) or later.
- Linux: Compatible with Ubuntu 18.04 LTS or later, Fedora 30 or later.

2. Storage Space:

- The app's installation package size is approximately 50 MB.

3. Permissions:

- Permission to install applications over USB and installation from unknown sources from 'Developer options '

4. Hardware Requirements:

- Android: The device should have at least 2GB of RAM and a minimum of 100 MB of available storage space.
- iOS: The app is compatible with iPhone 6s or later models.
- Windows: The device should have a minimum of 4GB of RAM and a dual-core processor.
- macOS: The app is compatible with Macs with Intel processors.
- Linux: The device should have a minimum of 2GB of RAM and a dual-core processor.

Logo:



Set up Flutter and Firebase:

1. Create a new Flutter project:

- Open your terminal or command prompt.
- Run the command `flutter create project_name` to create a new Flutter project. Replace "project_name" with the desired name for your project.

2. Add Firebase to your project:

- Visit the Firebase website (<https://firebase.google.com/>) and sign in with your Google account.
- Create a new Firebase project.
- Once your project is created, click on "Add app" and select the Flutter platform.
- Follow the instructions to register your app with Firebase.
- Download the `google-services.json` file, which contains your Firebase configuration.

3. Set up Firebase dependencies:

- Open your project in your chosen IDE.
- Open the `pubspec.yaml` file in the root of your Flutter project.
- Add the following dependencies:

dependencies:

`firebase_core: ^1.7.0`

`firebase_auth: ^3.1.0`

`cloud_firestore: ^3.1.0`

- Save the `pubspec.yaml` file.
- Run the command `flutter pub get` in the terminal to fetch the dependencies.

4. Configure Firebase in your Flutter project:

- Move the downloaded `google-services.json` file to the `android/app` directory of your Flutter project.
- Open the `android/build.gradle` file and add the following lines at the bottom of the file:
- `apply plugin: 'com.google.gms.google-services'`
- Open the `android/app/build.gradle` file and add the following line at the bottom of the file:
`gradle`
- `implementation 'com.google.firebase:firebase-core:20.0.0'`
- Save the `build.gradle` file.

5. Initialize Firebase in your Flutter app:

Open the main.dart file in the lib directory of your Flutter project.

- Import the necessary Firebase packages:
- `import 'package:firebase_core/firebase_core.dart';`
- Add the following code to initialize Firebase in the `main()` method:

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp();  
  runApp(MyApp());  
}
```

- Save the main.dart file.

Supported Platforms:

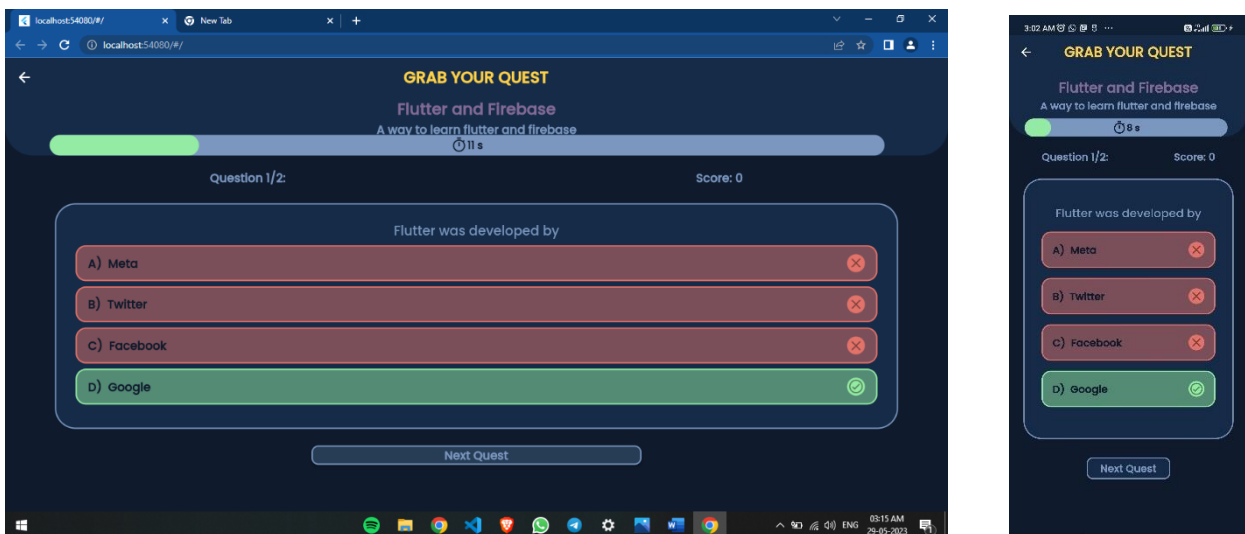


Fig1: Quest Runs in android , iOS and also as a web app

Since Flutter is a Cross Platform application development framework, higher level of flexibility is possible.

Database Design:

Firebase is deployed as the Backend database. Authentication, Fire store Database and Firebase Storage were used to create our project.

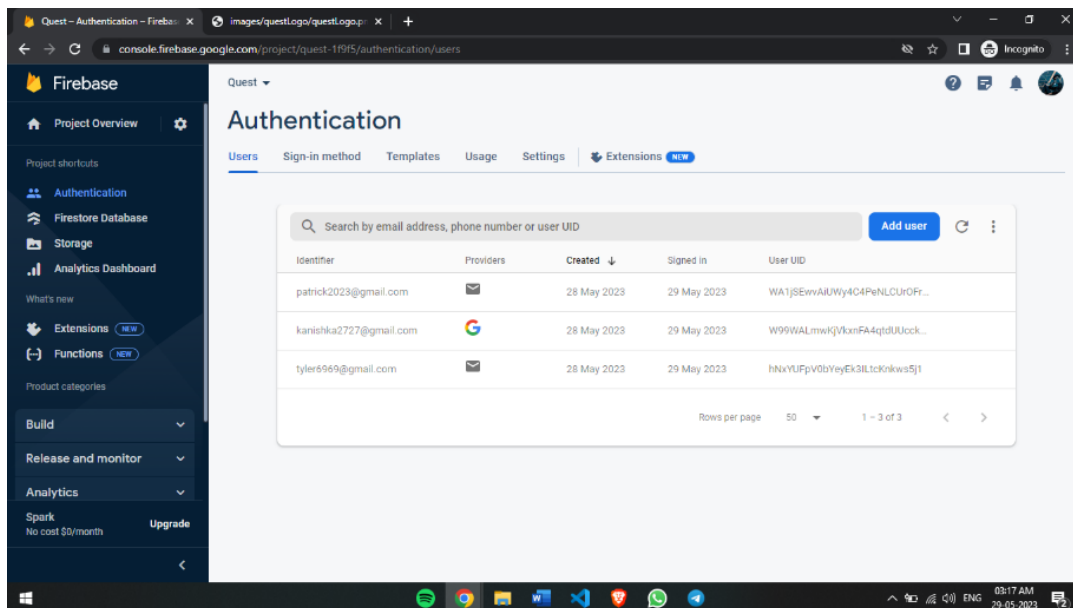


Fig1: Firebase Authentication

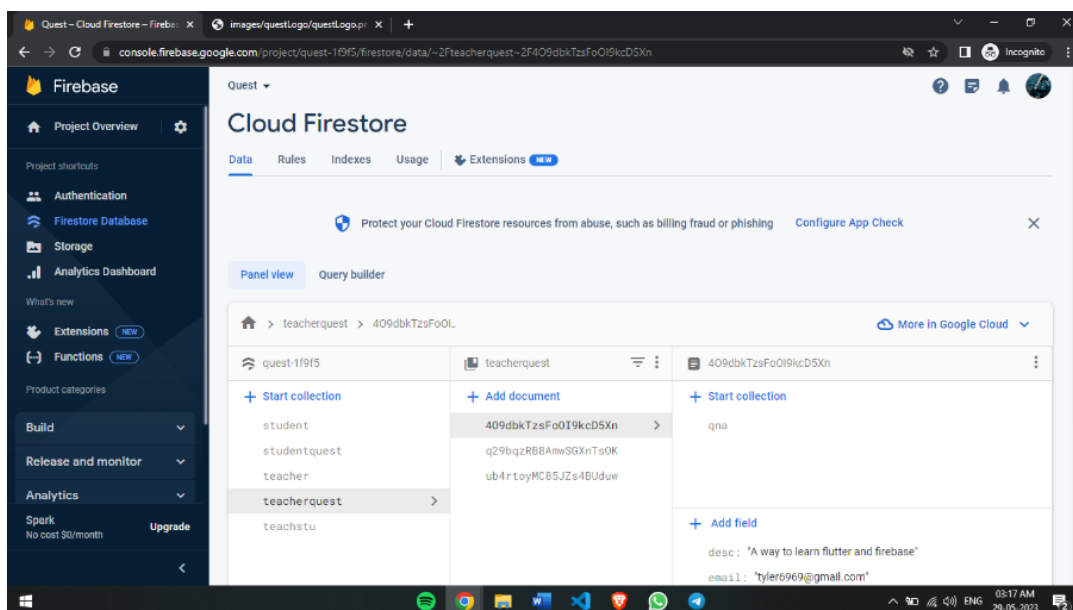


Fig2: Firebase Firestore Database

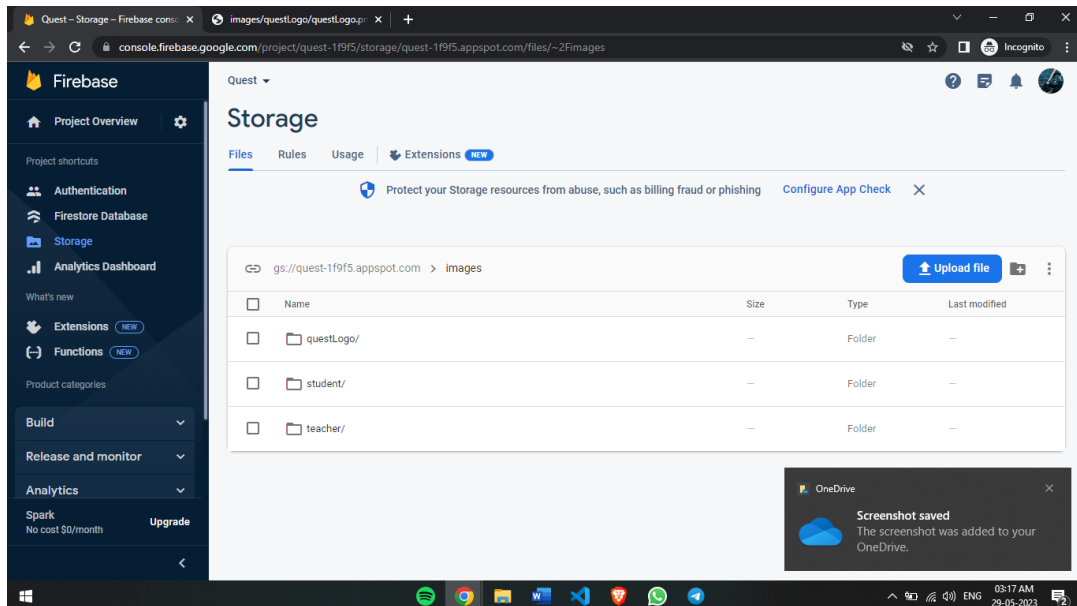


Fig3: Firebase Storage

Implementation:

On Boarding Screen:

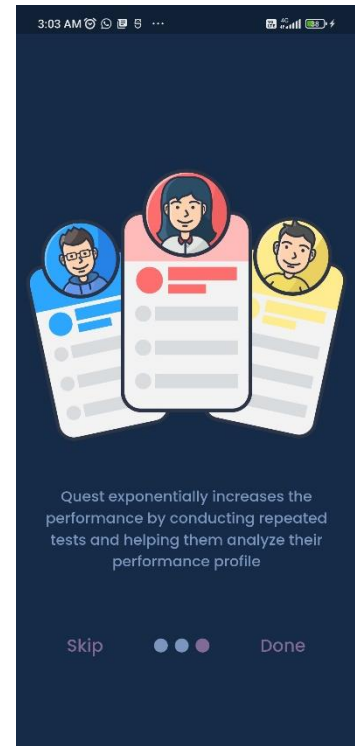
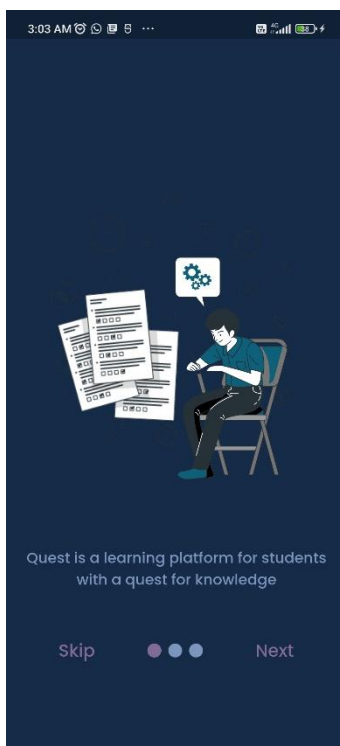


Fig4: On Boarding Screen while app launches

Login or Register Page:

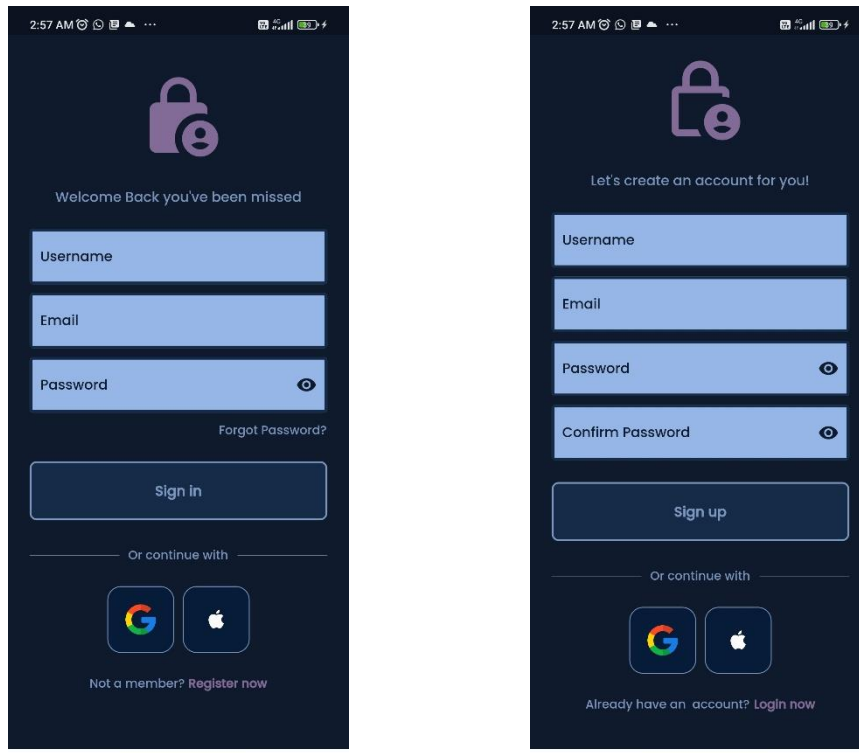


Fig5: Login and Register Page for Registering or Signing in the users

Teacher or Student Page:

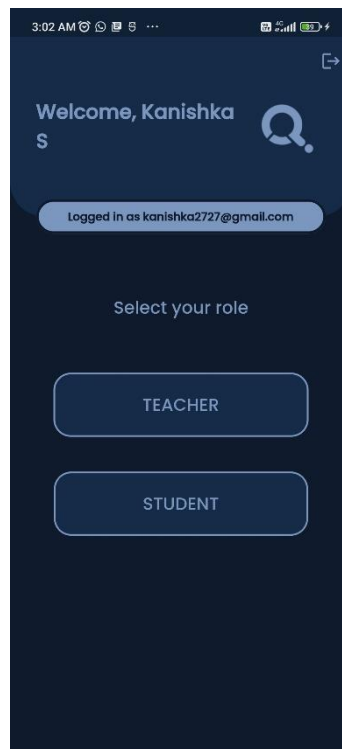


Fig6: Selection Page for Teacher or Student

Teacher Profile Page:

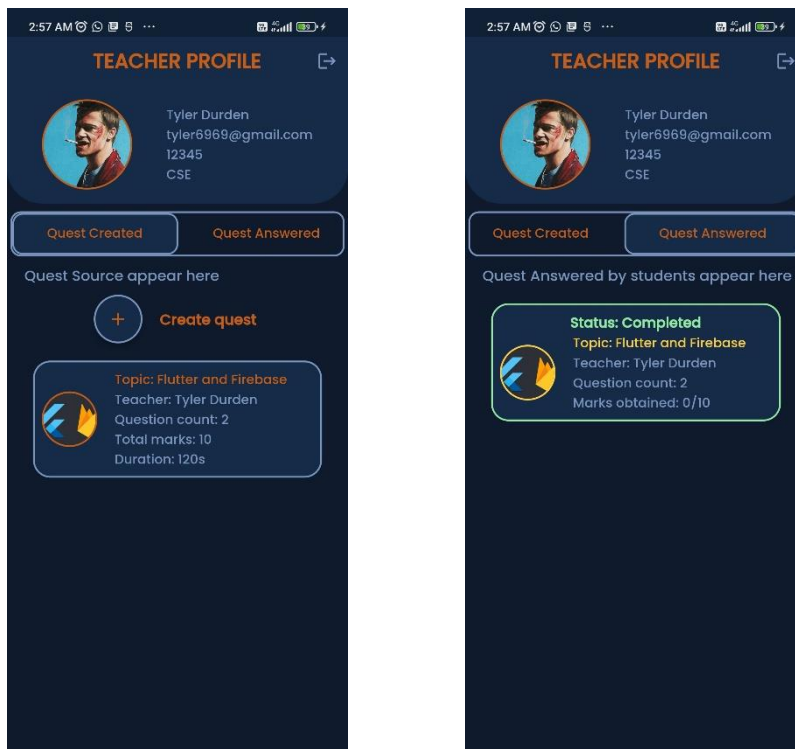


Fig7: Teacher Profile Page for Teacher Dashboard

Create Quest:

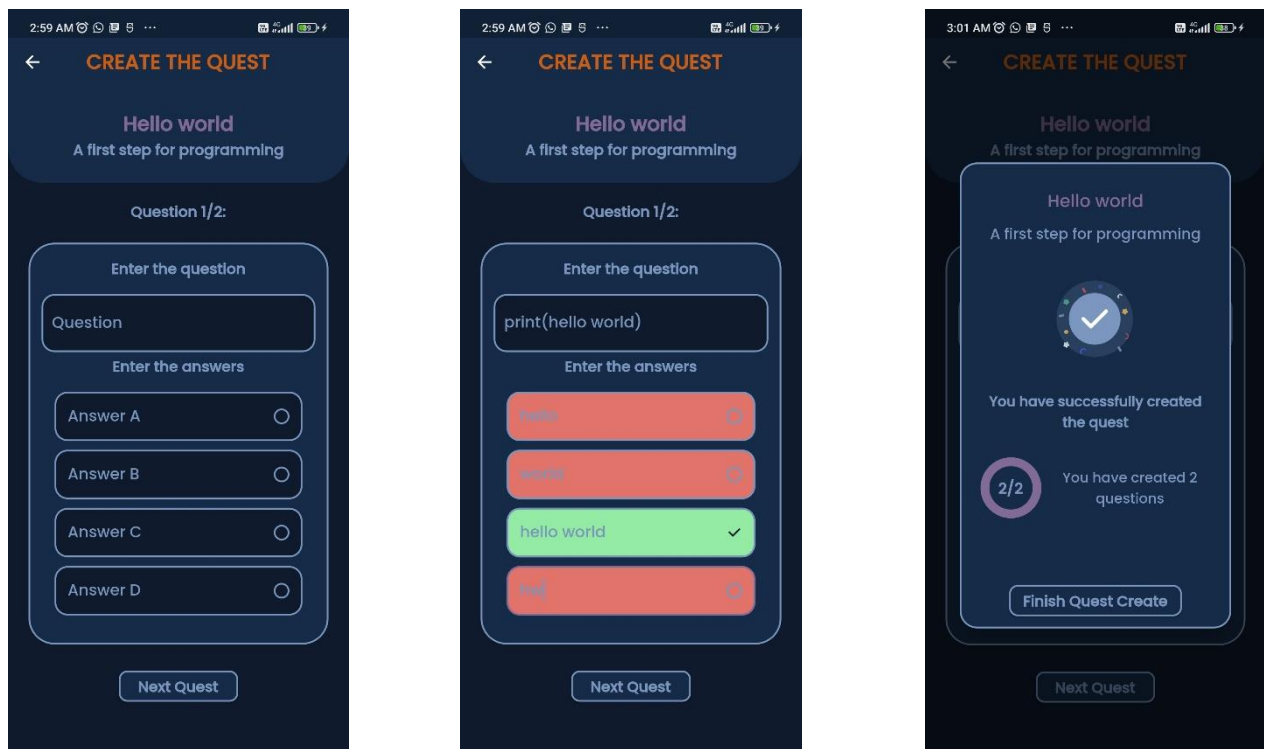


Fig8: Pages to create quiz for users

Student Profile Page:

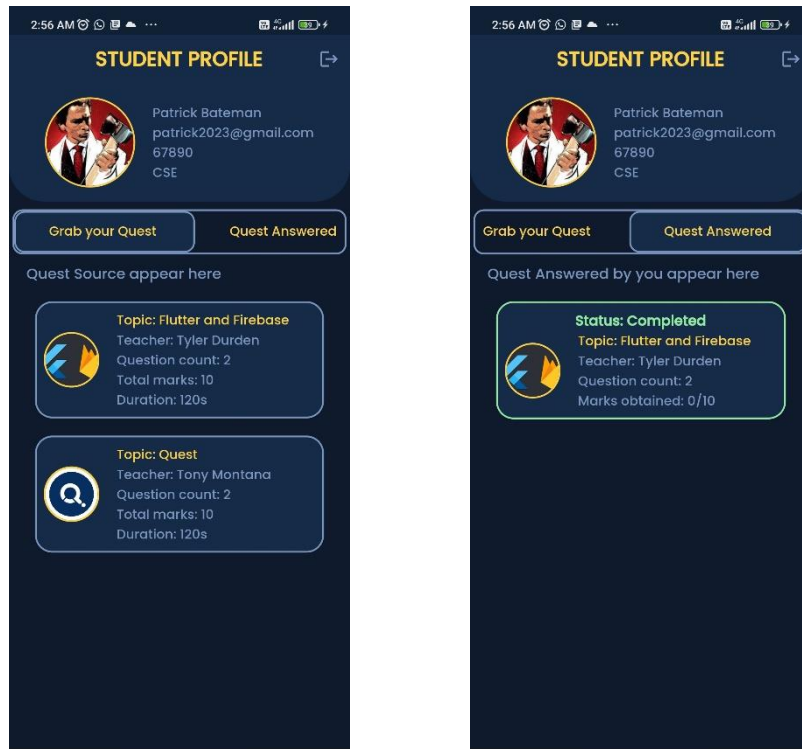


Fig9: Student Profile Page for student Dashboard

Answer Quest:

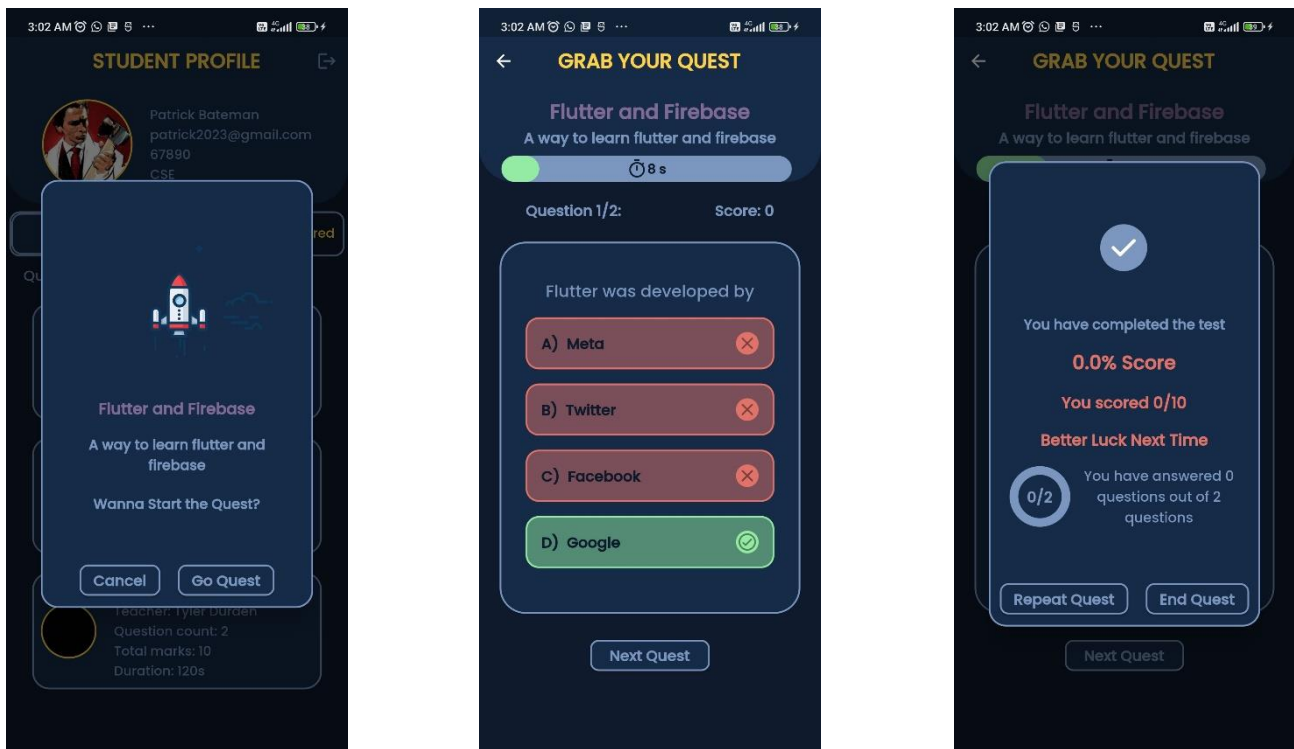


Fig10: Pages to Answer the quiz are created

Review Answers:

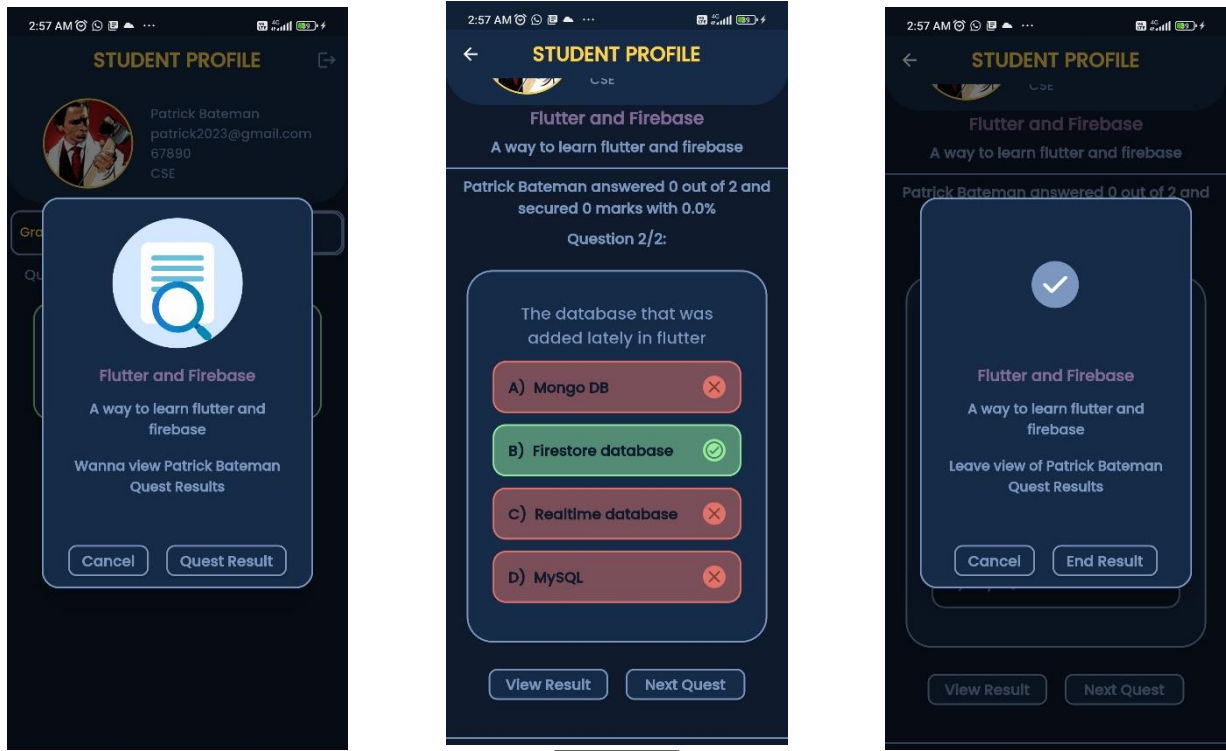


Fig11: Pages to Review the answered quiz