**Report: KnowHub (Developing a Simple Quiz App with Firebase in Flutter)**

**Developed By :** Zehra Khuwaja (20SW065) , Fiza Khuwaja (20SW063)

**Assigned By:** Maam Mariam (MAD)

**1. Introduction**

This report outlines the development of a simple quiz app using the Flutter framework with Firebase integration. The app will consist of multiple screens, including a loading page, quiz pages, correct/wrong answer feedback pages, and a final score page. Firebase Realtime Database will be used to store and retrieve quiz data.

**2. App Architecture**

The quiz app with Firebase integration will follow a similar architecture as previously described, with the addition of Firebase components:

1. **Loading Page:** Displayed while the app loads quiz data from Firebase.
2. **Quiz Pages**: Display quiz questions and answer choices to the user.
3. **Correct/Wrong Answer Pages**: Provide feedback on the user's answers.
4. **Score Page:** Display the final score.

**3. Functionality**

The app will have the following functionality:

1. **Loading Page:**

Display a loading indicator.

Fetch quiz data from Firebase Realtime Database.

Navigate to the first quiz question when data is loaded.

1. **Firebase Integration:**

Set up Firebase in the Flutter app, including authentication (if needed) and database configuration.

Store quiz questions, answer choices, correct answers, and user scores in the Firebase Realtime Database.

1. **Quiz Pages:**

Retrieve quiz questions and answer choices from Firebase.

Allow the user to select an answer.

Implement a timer for each question.

Automatically move to the next question after the timer expires or when the user selects an answer.

1. **Correct/Wrong Answer Pages:**

Display feedback indicating whether the user's answer was correct or wrong.

Show the correct answer if the user's answer was wrong.

Keep track of the user's score in Firebase.

1. **Score Page:**

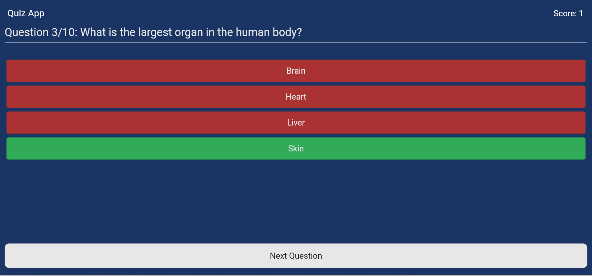
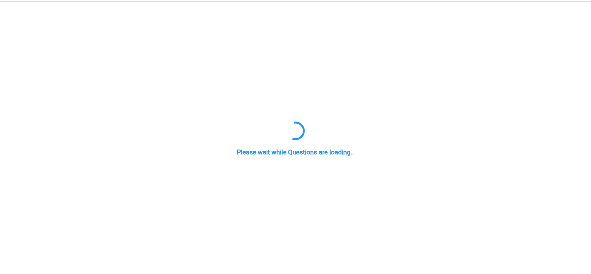
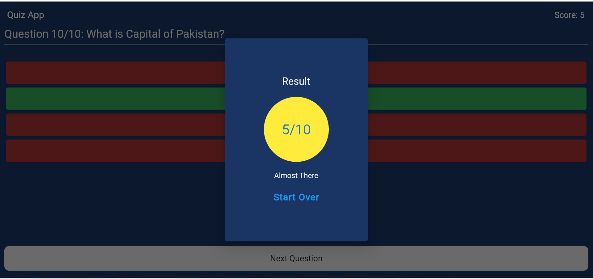
Retrieve the user's final score from Firebase.

Display the user's final score out of the total number of questions.

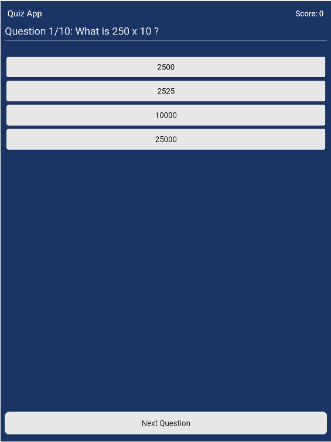
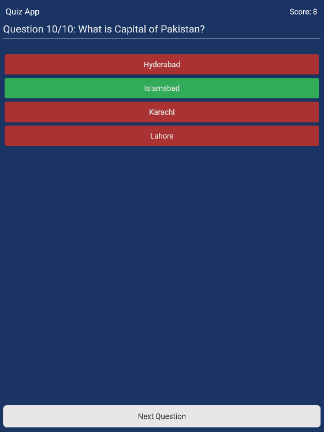
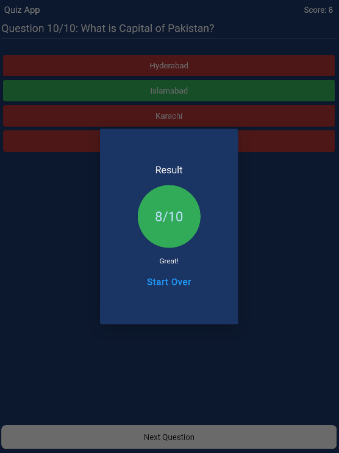
Allow the user to restart the quiz or exit the app.

**4.ScreenShots**

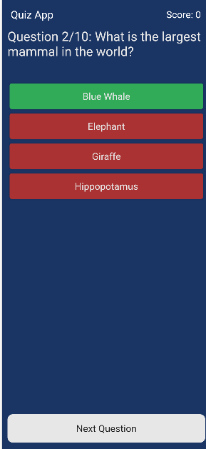
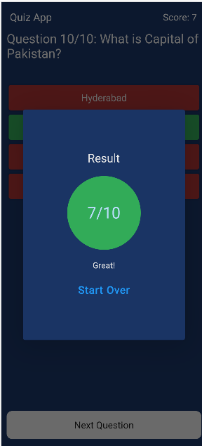
1. **DeskTop View**

** **

1. **Ipad View**

**  **

1. **Mobile View**

**  **

**5.Responsiveness**

The app's user interface will be designed to be responsive. This means that the app will adapt its layout and content to different screen sizes and orientations. Key strategies for achieving responsiveness include:

Using MediaQuery to adjust UI elements based on screen dimensions.

Employing layout widgets like Container, Row, Column, and Expanded to create flexible layouts.

Applying media queries to conditionally style elements based on screen size.

Ensuring that the app functions well in both portrait and landscape orientations.

Testing the app on various devices to verify responsiveness.

**6. App Development**

The app will be developed using the Flutter framework, Dart programming language, and Firebase for data storage and retrieval. Key Firebase components include Firebase Realtime Database

**7. Conclusion**

In conclusion, this report outlines the development of a simple quiz app in Flutter with Firebase integration. The app will use Firebase Realtime Database to store and retrieve quiz data, providing an interactive and engaging experience for users while testing their knowledge on various topics.

**8.Future Enhancements**

In the future, the app can be enhanced with additional features such as user authentication, the ability to create and share custom quizzes, and a leaderboard to compare scores with other users.

**9.GitHub Repository Link:**

**10. References**

* Flutter documentation: https://flutter.dev/
* Dart documentation: https://dart.dev/
* Firebase documentation: https://firebase.google.com/docs
* FlutterFire (Firebase plugins for Flutter): https://pub.dev/packages?q=flutterfire