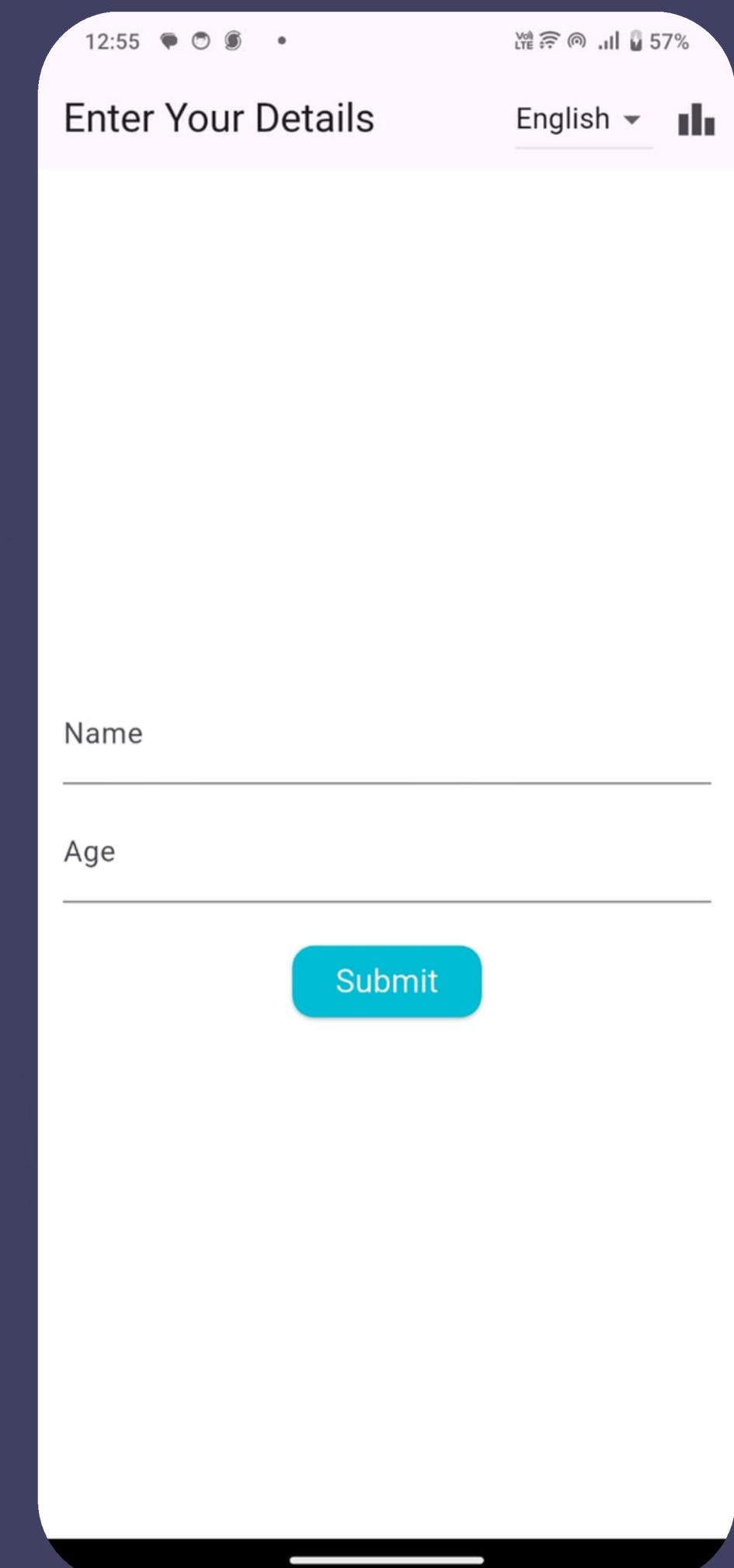


# IntelliQuiz: Unlock Your Cognitive Potential

IntelliQuiz is an innovative mobile app designed to measure both IQ and EQ. It offers a multilingual quiz experience with real-time leaderboard tracking. Get personalized cognitive insights to unlock your full potential.

by Aravindraj M



# Addressing the Problem Statement :

## Problem Statement

The basic idea is to create a app with quiz questions to calculate the IQ and EQ of the user. This quiz app needs to contain a series of iq and eq based questions to calculate intelligence of the user.

## Our Solution

IntelliQuiz provides an engaging mobile experience to provide a personalized, and accessible cognitive assessment. The app calculates the iq and eq points of the user and stores the data into the firestore database .

# Core Features for a Comprehensive



**Comprehensive  
Quizzes**



**Localization for  
getting user details**



**Data storage in cloud**



**Instant Score  
Calculation**

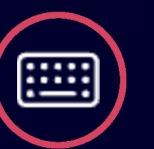
IntelliQuiz offers comprehensive IQ and EQ quizzes. It supports localization, app collects user data, and instantly calculates scores. Detailed performance analytics are also available.

# Localization: Bridging Language Barriers



## Seamless Language Switching

Effortlessly switch between English and Tamil within the app for getting the data from the user.



## Adaptive Input Forms

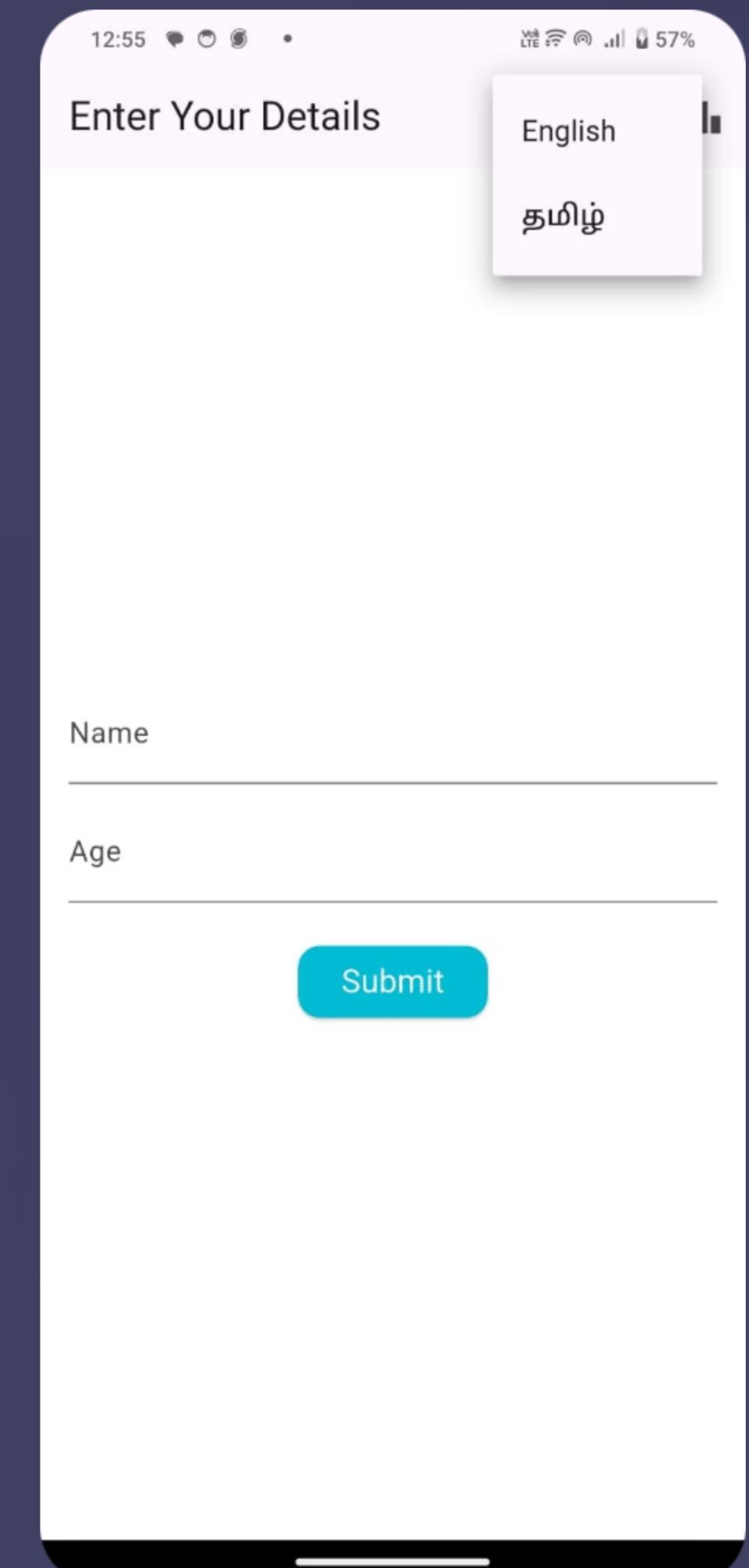
Customized input fields for name and age in both languages.

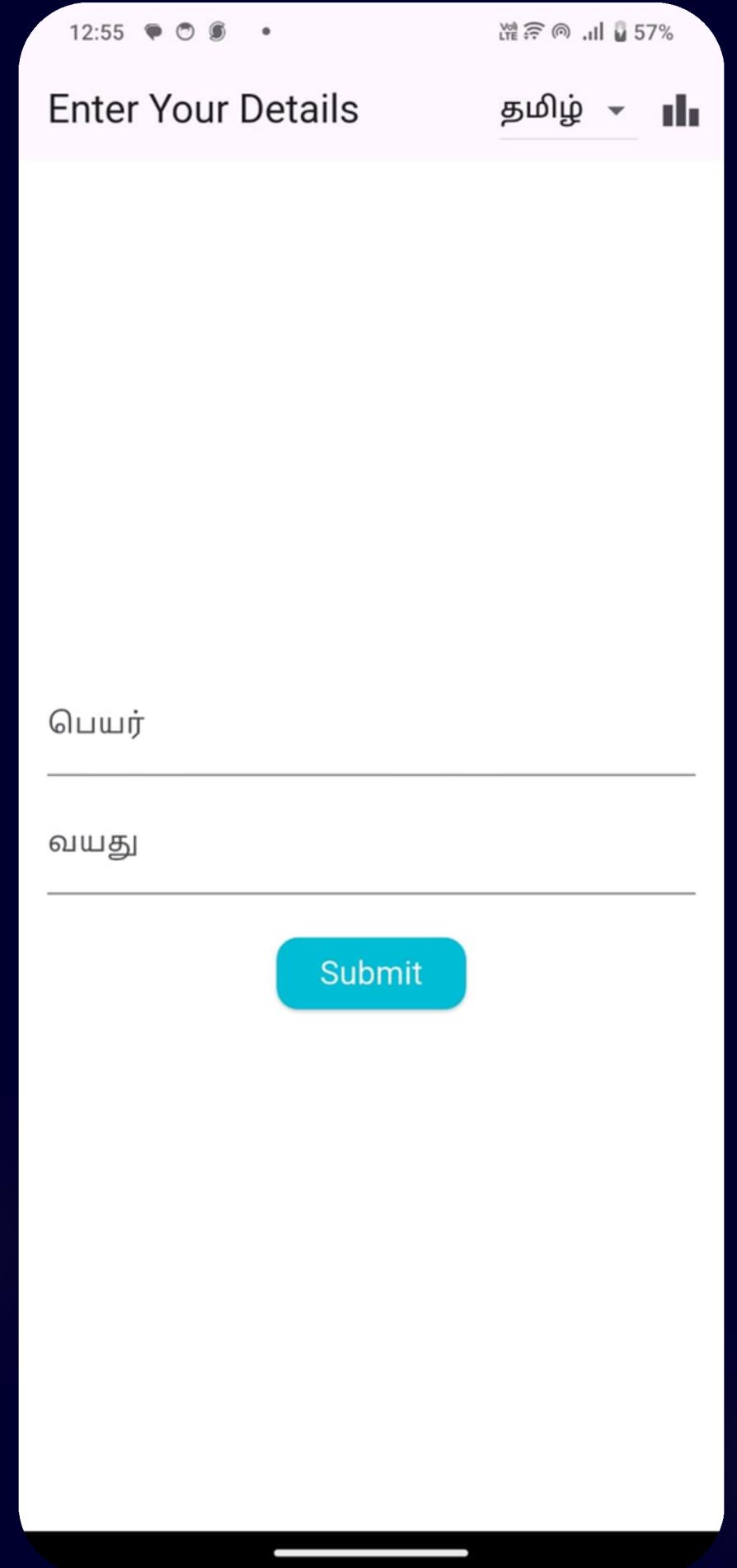


## Cultural Sensitivity

Culturally relevant question design ensures inclusivity and engagement.

IntelliQuiz ensures an inclusive user experience for diverse language speakers. Localization enables easy adaptation of the app. This allows it to better serve diverse user groups.





# Localization in Tamil

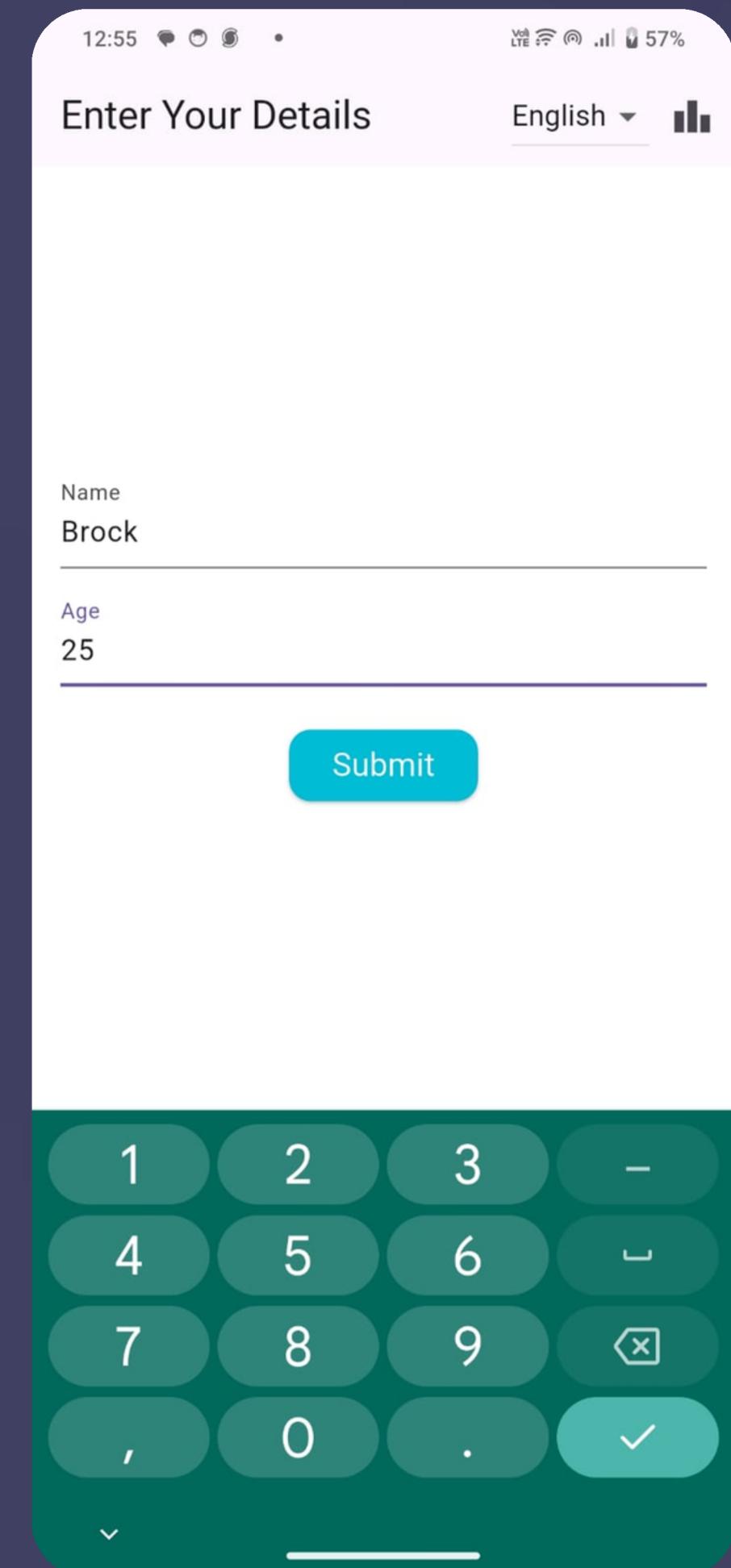
# Leaderboard Feature

The quiz app contains a leaderboard option which contains a leaderboard to mark the rank of the user with highest iq and eq score, the leaderboard gets updated frequently.



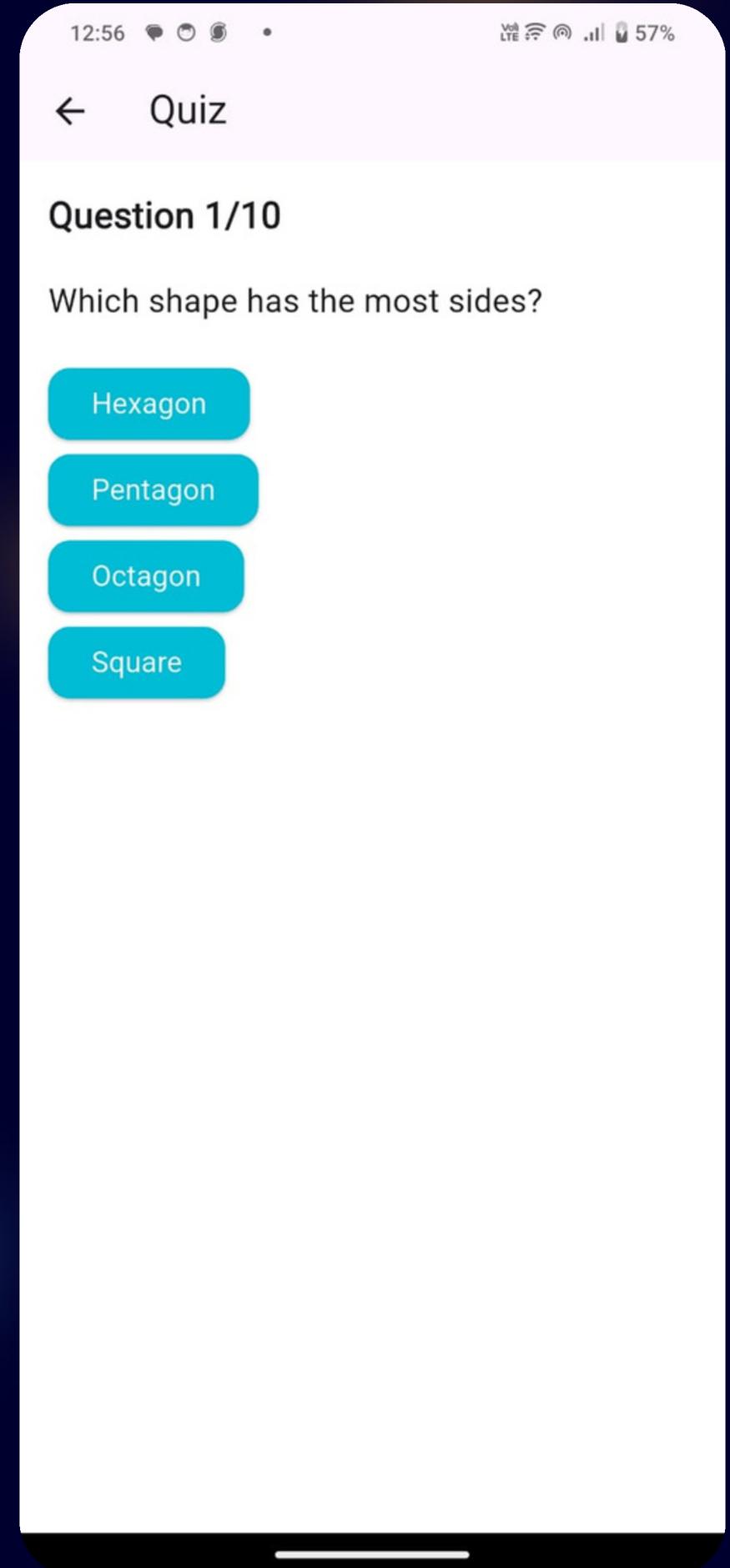
# Inputting User details.

The input fields gets the user name and age as an input and after calculation of the score they are stored in the firestore database.

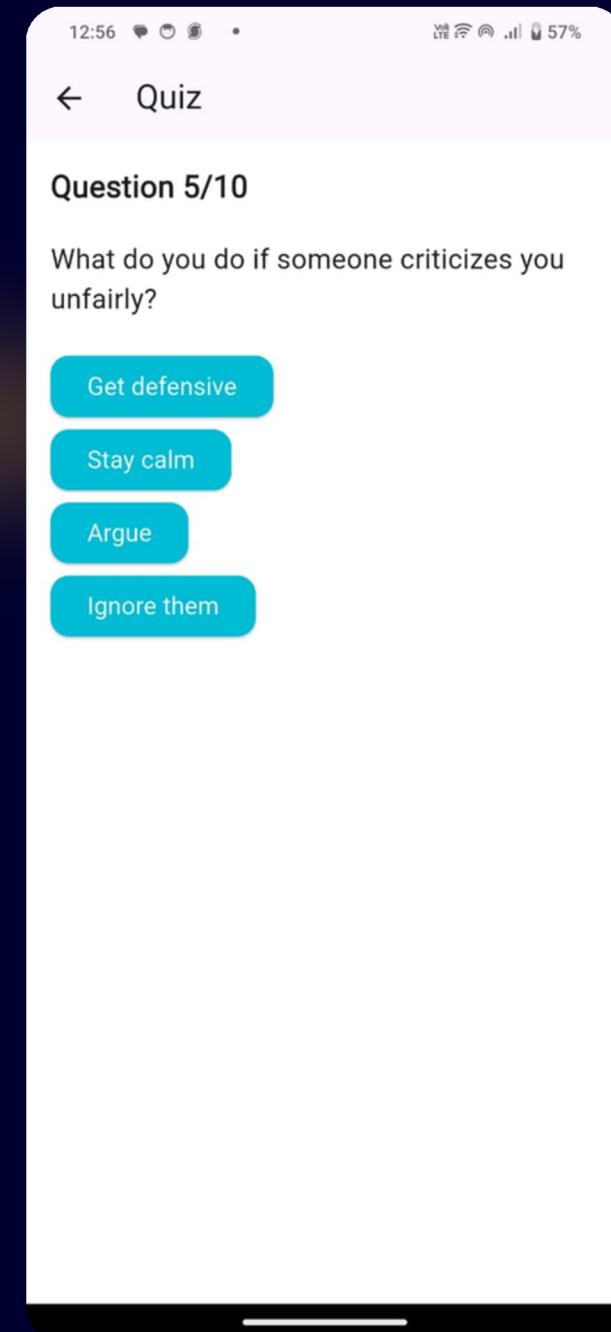
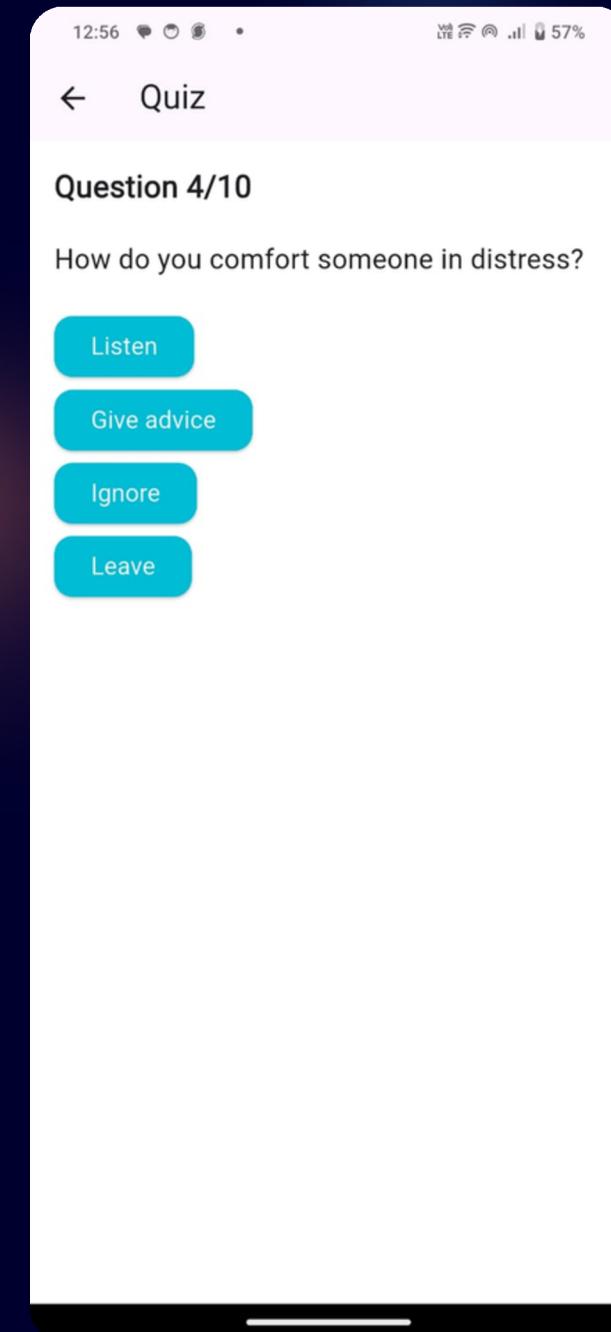
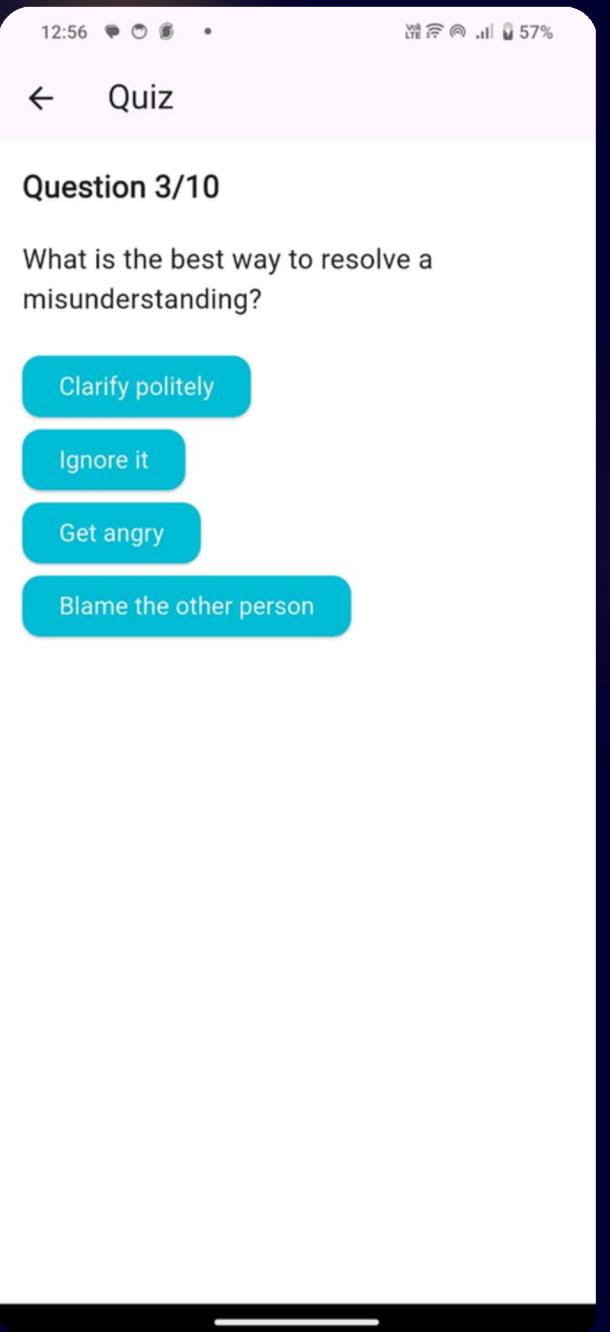
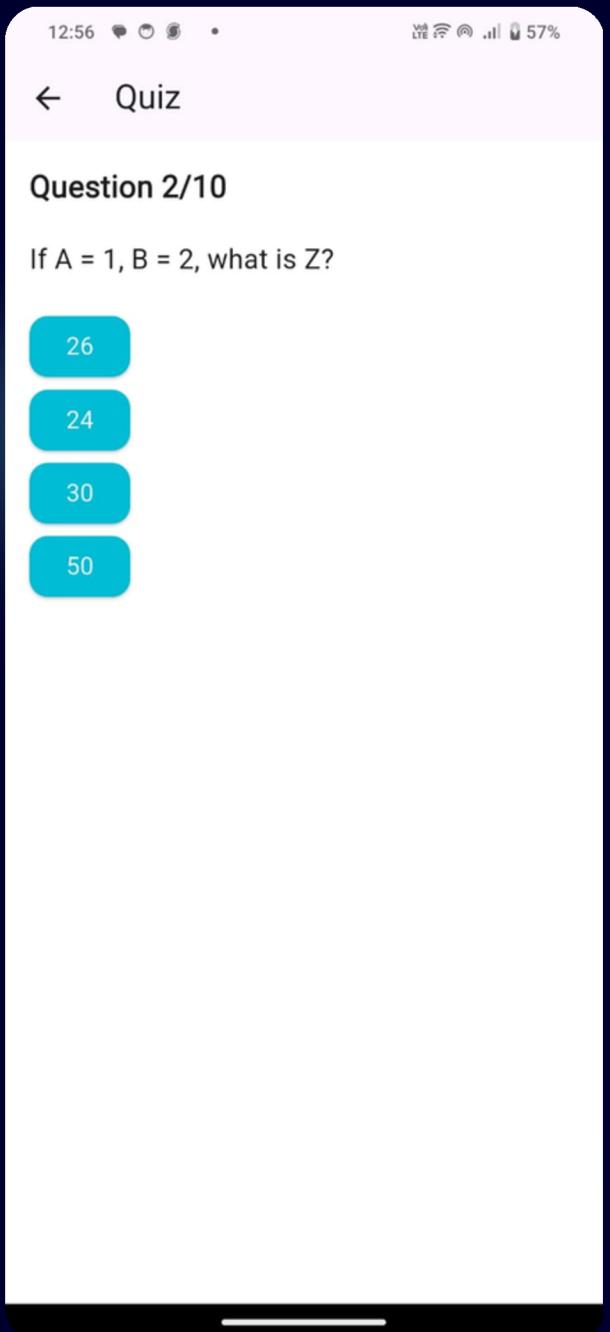


# Quiz Questions

A total of 10 questions will be asked by the app in a random order and the score will be calculated based on the answers.

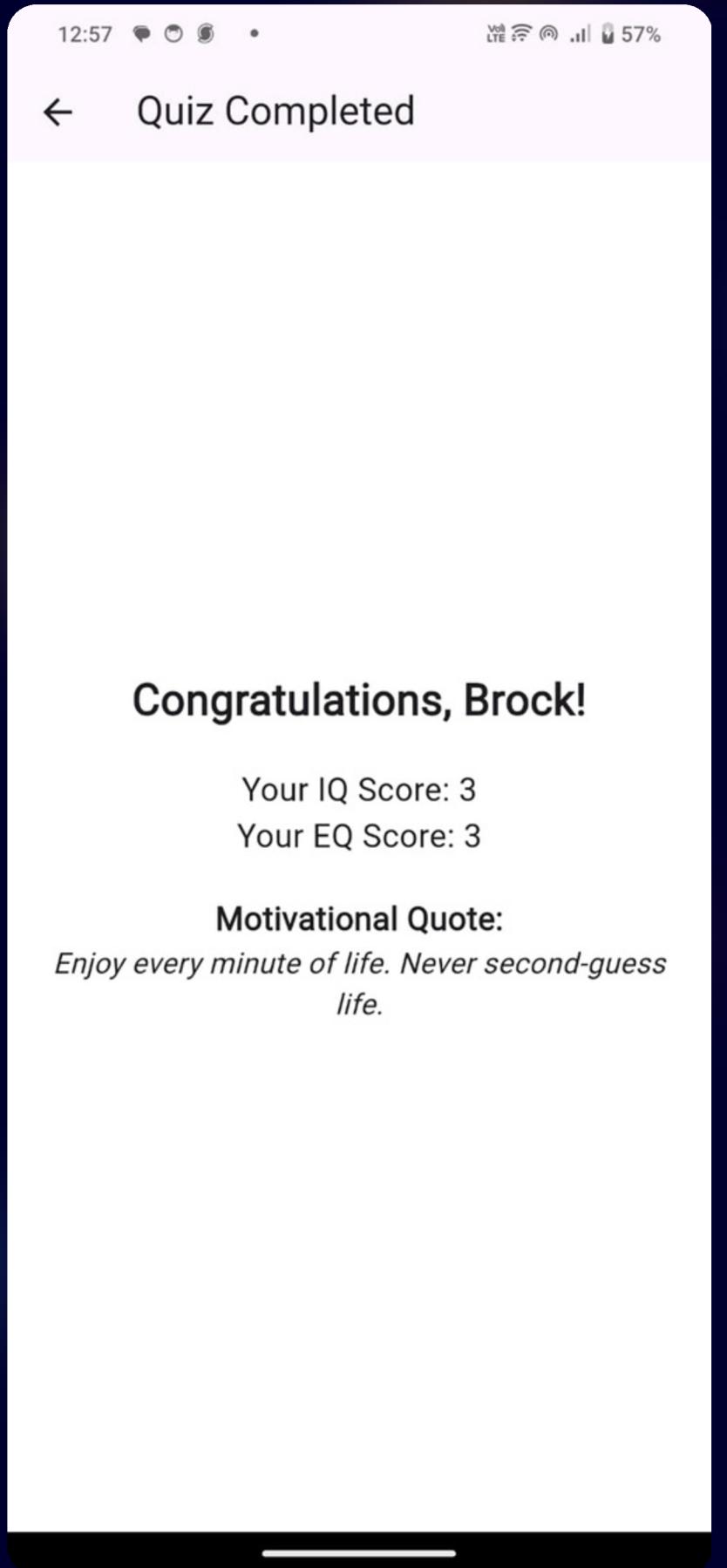


# Quiz Questions



# Quiz Completion Screen

Once the quiz is completed the IQ and EQ score is displayed to the user out of 5 total 10 questions 5 for eq and 5 for iq, along with the iq eq score a random motivational message is being displayed in score screen through a simple api call.



# Firebase: Firestore Database

The screenshot shows the Firebase Firestore Database interface. On the left, a sidebar menu includes 'Project Overview', 'Project shortcuts', 'Firestore Database' (which is selected and highlighted in blue), 'What's new', 'Genkit (NEW)', 'Product categories', 'Build', 'Run', 'Analytics', 'AI', 'All products', 'Related development tools' (with 'IDX' and 'Spark' options), and 'Upgrade' (No-cost (\$0/month)). The main area displays a hierarchical database structure under 'users'. At the top level, there is a collection named '(default)' containing a single document named 'users'. This document has three fields: 'age' (value: "25"), 'eqScore' (value: 3), and 'iqScore' (value: 3). It also contains a field 'name' with the value "Brock". The URL in the browser bar is 'https://console.firebase.google.com/project/\_/database/firestore/documents/users/WpBwB63AvKy7EjACvC89'.

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing    Configure App Check

Project Overview    X

Project shortcuts

Firestore Database

What's new

Genkit (NEW)

Product categories

Build

Run

Analytics

AI

All products

Related development tools

IDX ?

Spark No-cost (\$0/month)    Upgrade (NEW)

Panel view    Query builder    :

More in Google Cloud ▾

(default)

+ Start collection

users

+ Add document

KHFV3hRtAGTj3kNgc8Yd

WpBwB63AvKy7EjACvC89

daEed8HrWR3Jmwtgt506

+ Start collection

+ Add field

age: "25"

eqScore: 3

iqScore: 3

name: "Brock"

# Pushing to the database

```
    }

    void _submitScoresToFirebase() {
        FirebaseFirestore.instance.collection('users').add({
            'name': widget.name,
            'age': widget.age,
            'iqScore': _iqScore,
            'eqScore': _eqScore,
        });
    }
}
```

Once the iq and eq score is calculated all the details collected from the user along with the score is send to the database under users.

# Fetching data for the leaderboard

```
appbar: AppBar<Widget> build(BuildContext context) {
  return AppBar(
    title: Text('Leaderboard'),
    body: StreamBuilder(
      stream: FirebaseFirestore.instance
        .collection('users')
        .orderBy('iqScore', descending: true) // Sorting by IQ Score
        .limit(10) // Show top 10 users
        .snapshots(),
      builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Center(child: CircularProgressIndicator());
        }
        if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
          return Center(child: Text('No data available'));
        }
      },
    ),
  );
}
```

The data present in the leaderboaed are then fetched from the firestore database. The leaderboard is ordered based on the iq score

# API call for motivational quote

```
Future<void> _fetchMotivationalQuote() async {
    final response = await http.get(Uri.parse('https://zenquotes.io/api/random'));
    if (response.statusCode == 200) {
        setState(() {
            _quote = jsonDecode(response.body)[0]['q'];
        });
    }
}
```

# Stateful Widget :

```
class UserDetailsScreen extends StatefulWidget {
  @override
  _UserDetailsScreenState createState() => _UserDetailsScreenState();
}

class _UserDetailsScreenState extends State<UserDetailsScreen> {
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _ageController = TextEditingController();
  Locale _selectedLocale = Locale('en', '');

  void _changeLanguage(Locale locale) {
    setState(() {
      _selectedLocale = locale;
    });
  }
}
```

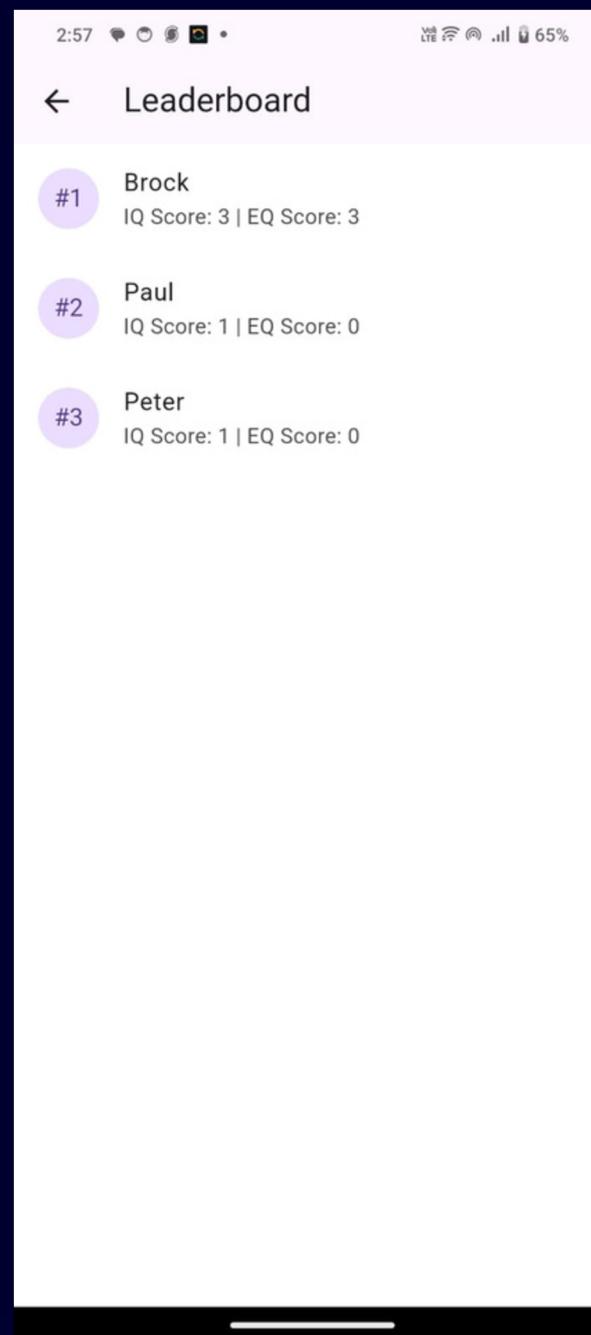
The user details are collected through this userdetailsscreen stateful widget, this widget is responsive to localization

# Stateless Widget :

```
class LeaderboardScreen extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(title: Text('Leaderboard')),  
            body: StreamBuilder(  
                stream: FirebaseFirestore.instance  
                    .collection('users')  
                    .orderBy('iqScore', descending: true) // Sorting by IQ Score  
                    .limit(10) // Show top 10 users  
                    .snapshots(),  
                builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {  
                    if (snapshot.connectionState == ConnectionState.waiting) {  
                        return Center(child: CircularProgressIndicator());  
                    }  
                    if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {  
                        return Center(child: Text('No data available'));  
                    }  
                },  
            ),  
        );  
    }  
}
```

The leaderboard details are displayed in a  
stateless widget

# Updated leaderboard :



The IQ EQ score of brock is being upated in the leaderboard