Without Safe Area

```
void main() {
  runApp(MaterialApp(
    home: Text("This is a Test",
      style: TextStyle(
        fontSize: 40,
        fontWeight: FontWeight.bold
      ),
  ),
  ),
  );
}
```

With Safe Area

```
void main() {
  runApp(MaterialApp(
    home: SafeArea(
      child: Text("This is a Test",
        style: TextStyle(
          fontSize: 40,
          fontWeight: FontWeight.bold
        ),
  ),
    ),
  ),
  );
}
```

Scaffold

```
void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text("Android",
        ),
      ),
      body: Center(
        child: Text("This is a sample Test",
        style: TextStyle(
          fontSize: 40,
        ),
        ),
      ),
    ),
  ),
  );
}
```

Row Axis

```dart
void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text("hello"),
      ),
      body: Rowpage(),
    ),
  ));
}


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

  @override
  State<Rowpage> createState() => _RowpageState();
}

class _RowpageState extends State<Rowpage> {
  @override
  Widget build(BuildContext context) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            Container(
              height: 100,
              width: 100,
              color: Colors.blue,
              child: Center(
                child: Text("First Child"),
              ),
            ),
          ],
        ),

        Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              height: 100,
              width: 100,
              color: Colors.green,
              child: Center(
                child: Text("First Child"),
              ),
            ),
          ],
        ),
        Column(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            Container(
              height: 100,
              width: 100,
              color: Colors.red,
              child: Center(
                child: Text("First Child"),
```

```
              ),
            ),
          ],
        ),
      ],
    );
  }
}
```

Dice

```
void main() {
  runApp(MaterialApp(
      home:Scaffold(
        appBar: AppBar(
          title: Text("Hello"
          ),
        ),
        body: Diecepage(),
      )
  ));
}
class Diecepage extends StatefulWidget {
  const Diecepage({super.key});

  @override
  State<Diecepage> createState() => _DiecepageState();
}

class _DiecepageState extends State<Diecepage> {
  int left=1;
  int right=2;
  void change(){
    setState(() {
      left=Random().nextInt(6)+1;
      right=Random().nextInt(6)+1;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Container(
        child: Row(
          children: [
            Expanded(
              flex: 1,
              child: TextButton(
                style: TextButton.styleFrom(
                  backgroundColor: Colors.blue
                ),
                child: Image.asset('images/dice$left.png'),
                onPressed: (){
                  change();
                },
              ),
            ),
```

```
            Expanded(
              flex: 1,
              child: TextButton(
                style: TextButton.styleFrom(
                    backgroundColor: Colors.blue
                ),
                child: Image.asset('images/dice$right.png'),
                onPressed: (){
                  change();
                },
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Xylo Phone

```
import 'package:assets_audio_player/assets_audio_player.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('XYLOPHONE'),
        centerTitle: true,
      ),
      body: XyloPage(),
    ),
  ));
}

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

  @override
  State<XyloPage> createState() => _XyloPageState();
}

class _XyloPageState extends State<XyloPage> {
  void playSound(int noteNumber){
    AssetsAudioPlayer.newPlayer().open(
      Audio('assets/note$noteNumber.wav'),
      autoStart: true,
      showNotification: true,
    );
  }
  @override
  Widget build(BuildContext context) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Expanded(child: TextButton(
          style: TextButton.styleFrom(
              backgroundColor: Colors.amberAccent
```

```
          ),
          onPressed: (){
            playSound(1);
          },
          child: Text('first'),
        ),
        ),
        Expanded(child: TextButton(
          style: TextButton.styleFrom(
              backgroundColor: Colors.brown
          ),
          onPressed: (){
            playSound(2);
          },
          child: Text('second'),
        ),
        ),
        Expanded(child: TextButton(
          style: TextButton.styleFrom(
              backgroundColor: Colors.yellow
          ),
          onPressed: (){
            playSound(3);
          },
          child: Text('third'),
        ),
        ),
        Expanded(child: TextButton(
          style: TextButton.styleFrom(
              backgroundColor: Colors.green
          ),
          onPressed: (){
            playSound(4);
          },
          child: Text('four'),
        ),
        ),
      ],
    );
  }
}
```

Quiz App

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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: Text('QUIZ'), centerTitle: true,),
      body: QuizPage(),
    ),
  ));
}

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

  @override
  State<QuizPage> createState() => _QuizPageState();
```

```
}
class Question{
  final String qText ;
  final bool qAns;
  Question({required this.qText, required this.qAns});
}

class Questions{
  List<Question> questionBank = [
    Question(qText: "Question1", qAns: true),
    Question(qText: "Question2", qAns: false),
    Question(qText: "Question3", qAns: true),

  ];
}
class _QuizPageState extends State<QuizPage> {
  int questionNumber = 0; // stores the questionNumber, default = first
  int currentScore = 0; // final score initialized to 0
  Questions questions = Questions(); // Create an object of the Question
Class

  void updateQuestionNumber() {
    setState(() {
      questionNumber = questionNumber + 1; // Increments the Question
Number
      print('$questionNumber');
    });
  }
  bool checkquestionNumber(int questionNumber) {
    return questionNumber < questions.questionBank.length ? true : false;
  }
  void udpateCurrentScore(bool choice, int question_number) {
// based on the choice (T/F button) score will be updated
    if (questions.questionBank.length == question_number) {
      print("End of questions");
    } else {
// checks the current user input against the list answer, if true
increments the count
      if (questions.questionBank[question_number].qAns == choice) {
        setState(() {
          currentScore++;
        });
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [
          Padding(
            padding: const EdgeInsets.only(top: 10.0),
            child: Center(
              child: Text(
                checkquestionNumber(questionNumber)?
                questions.questionBank[questionNumber].
                qText.toString():"END",
                style: TextStyle(
                  fontSize: 25.0,
```

```
                ),),
            ),
          ),
          SizedBox(height: 60.0,),
          if(checkquestionNumber(questionNumber))
            ElevatedButton(
              child: Text('True'),
              onPressed: (){
                setState(() {
                  if(questionNumber == questions.questionBank.length){
                    print('End of Questions');
                  }else{
                    udpateCurrentScore(true,questionNumber) ;
                    updateQuestionNumber();
                  }
                });
              },
            ),
          SizedBox(height: 20.0,),
          if(checkquestionNumber(questionNumber))
            ElevatedButton(
              child: Text('False'),
              onPressed: (){
                setState(() {
                  if(questionNumber == questions.questionBank.length){
                    print('End of Questions');
                  }else{
                    udpateCurrentScore(false,questionNumber) ;
                    updateQuestionNumber();
                  }
                });
              },
            ),
          SizedBox(height: 100.0,),

          Padding(
            padding: EdgeInsets.all(30.0),
            child: Center(
              child: Text(
                'Your current Score',
                style: TextStyle(fontSize: 30.0),
              ),
            ),
          ),
          Padding(
            padding: EdgeInsets.all(30.0),
            child: Center(
              child: Text(
                '$currentScore',
                style: TextStyle(fontSize: 30.0),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```