

```

import 'package:flutter/material.dart';
import 'package:flutter_custom_clippers/flutter_custom_clippers.dart';
import 'package:auto_size_text/auto_size_text.dart';
import 'package:news_app/category.dart';
import 'package:news_app/quizoption.dart';

```

```

class HomePage extends StatelessWidget {
  final List<Color> tileColors = [
    Colors.green,
    Colors.blue,
    Colors.purple,
    Colors.pink,
    Colors.indigo,
    Colors.lightBlue,
    Colors.amber,
    Colors.deepOrange,
    Colors.red,
    Colors.brown
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Quiz way'),
        elevation: 0,
      ),
      body: Stack(
        children: <Widget>[
          ClipPath(
            clipper: WaveClipperTwo(),
            child: Container(
              decoration:
                BoxDecoration(color: Theme.of(context).primaryColor),
              height: 200,
            ),
          ),
          CustomScrollView(
            physics: BouncingScrollPhysics(),
            slivers: <Widget>[
              SliverToBoxAdapter(
                child: Padding(
                  padding: const EdgeInsets.symmetric(
                    horizontal: 16.0, vertical: 8.0),
                  child: Text(
                    "Select a category to start the quiz",
                    style: TextStyle(
                      color: Colors.white,
                      fontWeight: FontWeight.w500,
                      fontSize: 16.0),
                  ),
                ),
              ),
              SliverPadding(
                padding: const EdgeInsets.all(16.0),
                sliver: SliverGrid(
                  gridDelegate:
SliverGridDelegateWithFixedCrossAxisCount(

```

```

        crossAxisCount:
MediaQuery.of(context).size.width >
        1000
        ? 7
        : MediaQuery.of(context).size.width > 600 ?
5 : 3,
        childAspectRatio: 1.2,
        crossAxisSpacing: 10.0,
        mainAxisSpacing: 10.0),
    delegate: SliverChildBuilderDelegate(
      _buildCategoryItem,
      childCount: categories.length,
    )),
  ),
),
),
),
));
}

```

```

Widget _buildCategoryItem(BuildContext context, int index) {
  Category category = categories[index];
  return MaterialButton(
    elevation: 1.0,
    highlightElevation: 1.0,
    onPressed: () => _categoryPressed(context, category),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10.0),
    ),
    color: Colors.grey.shade800,
    textColor: Colors.white70,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        if (category.icon != null) Icon(category.icon),
        if (category.icon != null) SizedBox(height: 5.0),
        AutoSizeText(
          category.name,
          minFontSize: 10.0,
          textAlign: TextAlign.center,
          maxLines: 3,
          wrapWords: false,
        ),
      ],
    ),
  );
}

```

```

_categoryPressed(BuildContext context, Category category) {
  showModalBottomSheet(
    context: context,
    builder: (sheetContext) => BottomSheet(
      builder: (_) => QuizOptionsDialog(
        category: category,
      ),
      onClosing: () {},
    ),
  );
}
}

```

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Quiz App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
        accentColor: Colors.indigo,
        fontFamily: "Montserrat",
        buttonColor: Colors.pink,
        buttonTheme: ButtonThemeData(
          buttonColor: Colors.pink,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(20.0),
          ),
          textTheme: ButtonTextTheme.primary
        ),
      ),
      home: HomePage(),
    );
  }
}
```

```

import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class Category{
  final int id;
  final String name;
  final dynamic icon;
  Category(this.id, this.name, {this.icon});
}

final List<Category> categories = [
  Category(9,"General Knowledge", icon: FontAwesomeIcons.globeAsia),
  Category(10,"Books", icon: FontAwesomeIcons.bookOpen),
  Category(11,"Film", icon: FontAwesomeIcons.video),
  Category(12,"Music", icon: FontAwesomeIcons.music),
  Category(13,"Musicals & Theatres", icon:
FontAwesomeIcons.theaterMasks),
  Category(14,"Television", icon: FontAwesomeIcons.tv),
  Category(15,"Video Games", icon: FontAwesomeIcons.gamepad),
  Category(16,"Board Games", icon: FontAwesomeIcons.chessBoard),
  Category(17,"Science & Nature", icon: FontAwesomeIcons.microscope),
  Category(18,"Computer", icon: FontAwesomeIcons.laptopCode),
  Category(19,"Maths", icon: FontAwesomeIcons.sortNumericDown),
  Category(20,"Mythology"),
  Category(21,"Sports", icon: FontAwesomeIcons.footballBall),
  Category(22,"Geography", icon: FontAwesomeIcons.mountain),
  Category(23,"History", icon: FontAwesomeIcons.monument),
  Category(24,"Politics"),
  Category(25,"Art", icon: FontAwesomeIcons.paintBrush),
  Category(26,"Celebrities"),
  Category(27,"Animals", icon: FontAwesomeIcons.dog),
  Category(28,"Vehicles", icon: FontAwesomeIcons.carAlt),
  Category(29,"Comics"),
  Category(30,"Gadgets", icon: FontAwesomeIcons.mobileAlt),
  Category(31,"Japanese Anime & Manga"),
  Category(32,"Cartoon & Animation"),
];

```

```
import 'dart:convert';

import 'package:news_app/category.dart';
import 'package:news_app/question.dart';
import 'package:http/http.dart' as http;

const String baseUrl = "https://opentdb.com/api.php";

Future<List<Question>> getQuestions(Category category, int total, String
difficulty) async {
  String url = "$baseUrl?amount=$total&category=${category.id}";
  if(difficulty != null) {
    url = "$url&difficulty=$difficulty";
  }
  http.Response res = await http.get(url);
  List<Map<String, dynamic>> questions =
List<Map<String, dynamic>>.from(json.decode(res.body) ["results"]);
  return Question.fromData(questions);
}
```

```

import 'package:flutter/material.dart';
import 'package:flutter_custom_clippers/flutter_custom_clippers.dart';
import 'package:html_unescape/html_unescape_small.dart';
import 'package:news_app/question.dart';
class CheckAnswersPage extends StatelessWidget {
  final List<Question> questions;
  final Map<int,dynamic> answers;

  const CheckAnswersPage({Key key, @required this.questions, @required
this.answers}) : super(key: key);

  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(
        title: Text('Check Answers'),
        elevation: 0,
      ),
      body: Stack(
        children: <Widget>[
          ClipPath(
            clipper: WaveClipperTwo(),
            child: Container(
              decoration: BoxDecoration(
                color: Theme.of(context).primaryColor
              ),
              height: 200,
            ),
          ),
          ListView.builder(
            padding: const EdgeInsets.all(16.0),
            itemCount: questions.length+1,
            itemBuilder: _buildItem,

          )
        ],
      ),
    );
  }
  Widget _buildItem(BuildContext context, int index) {
    if(index == questions.length) {
      return RaisedButton(
        child: Text("Done"),
        onPressed: (){
          Navigator.of(context).popUntil (ModalRoute.withName (Navigator.defaultRoute
Name));
        },
      );
    }
    Question question = questions[index];
    bool correct = question.correctAnswer == answers[index];
    return Card(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Text(HtmlUnescape().convert(question.question), style:
TextStyle(

```

```

        color: Colors.black,
        fontWeight: FontWeight.w500,
        fontSize: 16.0
    ),),
    SizedBox(height: 5.0),
    Text(HtmlUnescape().convert("${answers[index]}"), style:
TextStyle(
        color: correct ? Colors.green : Colors.red,
        fontSize: 18.0,
        fontWeight: FontWeight.bold
    ),),
    SizedBox(height: 5.0),
    correct ? Container(): Text.rich(TextSpan(
        children: [
            TextSpan(text: "Answer: "),
            TextSpan(text:
HtmlUnescape().convert(question.correctAnswer) , style: TextStyle(
                fontWeight: FontWeight.w500
            ))
        ]
    ),style: TextStyle(
        fontSize: 16.0
    ),)
    ],
    ),
    ),
    );
}
}

```

```

import 'package:flutter/material.dart';

class ErrorPage extends StatelessWidget {
  final String message;

  const ErrorPage({Key key, this.message = "There was an unknown error."
}) : super(key: key);

  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(
        title: Text('Error'),
        elevation: 0,
      ),
      body: Container(
        height: double.infinity,
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Theme.of(context).primaryColor,
              Theme.of(context).accentColor
            ],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter
          )
        ),
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Center(
            child: Card(
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: <Widget>[
                    Text(message, textAlign: TextAlign.center, style:
TextStyle(
                      fontSize: 20.0,
                      fontWeight: FontWeight.w500,
                      color: Colors.red
                    ),),
                    SizedBox(height: 20.0),
                    RaisedButton(
                      child: Text("Try Again"),
                      onPressed: ()=> Navigator.pop(context),
                    )
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }
}

```



```

enum Type {
    multiple,
    boolean
}

enum Difficulty {
    easy,
    medium,
    hard
}

class Question {
    final String categoryName;
    final Type type;
    final Difficulty difficulty;
    final String question;
    final String correctAnswer;
    final List<dynamic> incorrectAnswers;

    Question({this.categoryName, this.type, this.difficulty, this.question,
this.correctAnswer, this.incorrectAnswers});

    Question.fromMap(Map<String, dynamic> data):
        categoryName = data["category"],
        type = data["type"] == "multiple" ? Type.multiple : Type.boolean,
        difficulty = data["difficulty"] == "easy" ? Difficulty.easy :
data["difficulty"] == "medium" ? Difficulty.medium : Difficulty.hard,
        question = data["question"],
        correctAnswer = data["correct_answer"],
        incorrectAnswers = data["incorrect_answers"];

    static List<Question> fromData(List<Map<String, dynamic>> data) {
        return data.map((question) => Question.fromMap(question)).toList();
    }
}

```

```

import 'package:flutter/material.dart';
import 'package:news_app/checkanswer.dart';
import 'package:news_app/question.dart';
class QuizFinishedPage extends StatelessWidget {
  final List<Question> questions;
  final Map<int, dynamic> answers;

  int correctAnswers;
  QuizFinishedPage({Key key, @required this.questions, @required
this.answers}): super(key: key) {

  }

  @override
  Widget build(BuildContext context){
    int correct = 0;
    this.answers.forEach((index,value){
      if(this.questions[index].correctAnswer == value)
        correct++;
    });
    final TextStyle titleStyle = TextStyle(
      color: Colors.black87,
      fontSize: 16.0,
      fontWeight: FontWeight.w500
    );
    final TextStyle trailingStyle = TextStyle(
      color: Theme.of(context).primaryColor,
      fontSize: 20.0,
      fontWeight: FontWeight.bold
    );

    return Scaffold(
      appBar: AppBar(
        title: Text('Result'),
        elevation: 0,
      ),
      body: Container(
        height: double.infinity,
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Theme.of(context).primaryColor,
              Theme.of(context).accentColor
            ],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter
          )
        ),
      ),
      child: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: <Widget>[
            Card(
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10.0)
              ),
              child: ListTile(
                contentPadding: const EdgeInsets.all(16.0),
                title: Text("Total Questions", style: titleStyle),

```

```

        trailing: Text("${questions.length}", style:
trailingStyle),
    ),
),
    SizedBox(height: 10.0),
    Card(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0)
        ),
        child: ListTile(
            contentPadding: const EdgeInsets.all(16.0),
            title: Text("Score", style: titleStyle),
            trailing: Text("${correct/questions.length * 100}%",
style: trailingStyle),
        ),
    ),
    SizedBox(height: 10.0),
    Card(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0)
        ),
        child: ListTile(
            contentPadding: const EdgeInsets.all(16.0),
            title: Text("Correct Answers", style: titleStyle),
            trailing: Text("$correct/${questions.length}", style:
trailingStyle),
        ),
    ),
    SizedBox(height: 10.0),
    Card(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0)
        ),
        child: ListTile(
            contentPadding: const EdgeInsets.all(16.0),
            title: Text("Incorrect Answers", style: titleStyle),
            trailing: Text("${questions.length -
correct}/${questions.length}", style: trailingStyle),
        ),
    ),
    SizedBox(height: 20.0),
    Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: <Widget>[
            RaisedButton(
                padding: const EdgeInsets.symmetric(horizontal: 16.0,
vertical: 20.0),
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10.0),
                ),
                color:
Theme.of(context).accentColor.withOpacity(0.8),
                child: Text("Goto Home"),
                onPressed: () => Navigator.pop(context),
            ),
            RaisedButton(
                padding: const EdgeInsets.symmetric(horizontal: 16.0,
vertical: 20.0),
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10.0),

```

```

        ),
        color: Theme.of(context).primaryColor,
        child: Text("Check Answers"),
        onPressed: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (_) => CheckAnswersPage(questions:
questions, answers: answers,)
            ));
        },
    ),
],
),
],
),
),
),
);
}
}

```

```

import 'dart:io';

import 'package:flutter/material.dart';
import 'package:news_app/api_provider.dart';
import 'package:news_app/category.dart';
import 'package:news_app/error.dart';
import 'package:news_app/question.dart';
import 'package:news_app/quizpage.dart';

class QuizOptionsDialog extends StatefulWidget {
  final Category category;

  const QuizOptionsDialog({Key key, this.category}) : super(key: key);

  @override
  _QuizOptionsDialogState createState() => _QuizOptionsDialogState();
}

class _QuizOptionsDialogState extends State<QuizOptionsDialog> {
  int _noOfQuestions;
  String _difficulty;
  bool processing;

  @override
  void initState() {
    super.initState();
    _noOfQuestions = 10;
    _difficulty = "easy";
    processing = false;
  }

  @override
  Widget build(BuildContext context){
    return SingleChildScrollView(
      child: Column(
        children: <Widget>[
          Container(
            width: double.infinity,
            padding: const EdgeInsets.all(16.0),
            color: Colors.grey.shade200,
            child: Text(widget.category.name, style:
Theme.of(context).textTheme.title.copyWith(),),
          ),
          SizedBox(height: 10.0),
          Text("Select Total Number of Questions"),
          SizedBox(
            width: double.infinity,
            child: Wrap(
              alignment: WrapAlignment.center,
              runAlignment: WrapAlignment.center,
              runSpacing: 16.0,
              spacing: 16.0,
              children: <Widget>[
                SizedBox(width: 0.0),
                ActionChip(
                  label: Text("10"),
                  labelStyle: TextStyle(color: Colors.white),
                  backgroundColor: _noOfQuestions == 10 ? Colors.indigo :
Colors.grey.shade600,

```

```

                onPressed: () => _selectNumberOfQuestions(10),
            ),
            ActionChip(
              label: Text("20"),
              labelStyle: TextStyle(color: Colors.white),
              backgroundColor: _noOfQuestions == 20 ? Colors.indigo :
Colors.grey.shade600,
              onPressed: () => _selectNumberOfQuestions(20),
            ),
            ActionChip(
              label: Text("30"),
              labelStyle: TextStyle(color: Colors.white),
              backgroundColor: _noOfQuestions == 30 ? Colors.indigo :
Colors.grey.shade600,
              onPressed: () => _selectNumberOfQuestions(30),
            ),
            ActionChip(
              label: Text("40"),
              labelStyle: TextStyle(color: Colors.white),
              backgroundColor: _noOfQuestions == 40 ? Colors.indigo :
Colors.grey.shade600,
              onPressed: () => _selectNumberOfQuestions(40),
            ),
            ActionChip(
              label: Text("50"),
              labelStyle: TextStyle(color: Colors.white),
              backgroundColor: _noOfQuestions == 50 ? Colors.indigo :
Colors.grey.shade600,
              onPressed: () => _selectNumberOfQuestions(50),
            ),
          ],
        ),
      ),
    SizedBox(height: 20.0),
    Text("Select Difficulty"),
    SizedBox(
      width: double.infinity,
      child: Wrap(
        alignment: WrapAlignment.center,
        runAlignment: WrapAlignment.center,
        runSpacing: 16.0,
        spacing: 16.0,
        children: <Widget>[
          SizedBox(width: 0.0),
          ActionChip(
            label: Text("Any"),
            labelStyle: TextStyle(color: Colors.white),
            backgroundColor: _difficulty == null ? Colors.indigo :
Colors.grey.shade600,
            onPressed: () => _selectDifficulty(null),
          ),
          ActionChip(
            label: Text("Easy"),
            labelStyle: TextStyle(color: Colors.white),
            backgroundColor: _difficulty == "easy" ? Colors.indigo
: Colors.grey.shade600,
            onPressed: () => _selectDifficulty("easy"),
          ),
          ActionChip(

```



```
        Navigator.pushReplacement(context, MaterialPageRoute(
            builder: (_) => ErrorPage(message: "Can't reach the servers, \n
Please check your internet connection.",)
        ));
    } catch(e){
        print(e.message);
        Navigator.pushReplacement(context, MaterialPageRoute(
            builder: (_) => ErrorPage(message: "Unexpected error trying to
connect to the API",)
        ));
    }
    setState(() {
        processing=false;
    });
}
}
```





```

        child: Text("\$_currentIndex + 1"),
      ),
      SizedBox(width: 16.0),
      Expanded(
        child: Text(
          HtmlUnescape().convert(
            widget.questions[_currentIndex].question),
          softWrap: true,
          style: MediaQuery.of(context).size.width > 800
            ? _questionStyle.copyWith(fontSize: 30.0)
            : _questionStyle,
        ),
      ),
    ],
  ),
  SizedBox(height: 20.0),
  Card(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: <Widget>[
        ...options.map((option) => RadioListTile(
          title:
Text(HtmlUnescape().convert("\$option")), style:
MediaQuery.of(context).size.width > 800
          ? TextStyle(
            fontSize: 30.0
          ) : null,)),
        groupValue: _answers[_currentIndex],
        value: option,
        onChanged: (value) {
          setState(() {
            _answers[_currentIndex] = option;
          });
        },
      ]),
    ),
  ),
  Expanded(
    child: Container(
      alignment: Alignment.bottomCenter,
      child: RaisedButton(
        padding: MediaQuery.of(context).size.width > 800
          ? const EdgeInsets.symmetric(vertical:
20.0, horizontal: 64.0) : null,
        child: Text(
          _currentIndex == (widget.questions.length - 1)
            ? "Submit"
            : "Next", style:
MediaQuery.of(context).size.width > 800
            ? TextStyle(fontSize: 30.0) : null,)),
        onPressed: _nextSubmit,
      ),
    ),
  ),
],
),
],

```

```

        ),
    ),
);
}

void _nextSubmit() {
    if (_answers[_currentIndex] == null) {
        _key.currentState.showSnackBar(SnackBar(
            content: Text("You must select an answer to continue."),
        ));
        return;
    }
    if (_currentIndex < (widget.questions.length - 1)) {
        setState(() {
            _currentIndex++;
        });
    } else {
        Navigator.of(context).pushReplacement(MaterialPageRoute(
            builder: (_) => QuizFinishedPage(
                questions: widget.questions, answers: _answers)));
    }
}

Future<bool> _onWillPop() async {
    return showDialog<bool>(
        context: context,
        builder: (_) {
            return AlertDialog(
                content: Text(
                    "Are you sure you want to quit the quiz? All your
progress will be lost."),
                title: Text("Warning!"),
                actions: <Widget>[
                    FlatButton(
                        child: Text("Yes"),
                        onPressed: () {
                            Navigator.pop(context, true);
                        },
                    ),
                    FlatButton(
                        child: Text("No"),
                        onPressed: () {
                            Navigator.pop(context, false);
                        },
                    ),
                ],
            );
        });
}
}

```