***Name: Amish Sabir***

***Reg No: FA19\_BCS\_090***

**welcome.dart**

*import* 'package:final\_quiz\_app/main.dart';  
*import* 'package:flutter/material.dart';  
*import* 'package:mailto/mailto.dart';  
*import* 'dart:async';  
*import* 'package:share\_plus/share\_plus.dart';  
*import* 'package:url\_launcher/url\_launcher.dart';  
*import* 'package:google\_fonts/google\_fonts.dart';  
*void* main() => runApp(SplashScreen());  
  
  
*class* SplashScreen *extends* StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
  
 *return* MaterialApp(  
 title: 'Splash Screen',  
 theme: ThemeData(  
 primarySwatch: Colors.*amber*,  
 ),  
 home: MyHomePage(),  
 debugShowCheckedModeBanner: *false*,  
 );  
 }  
}  
*class* MyHomePage *extends* StatefulWidget {  
 @override  
 \_MyHomePageState createState() => \_MyHomePageState();  
}  
*class* \_MyHomePageState *extends* State<MyHomePage> {  
 @override  
 *void* initState() {  
 *super*.initState();  
 Timer(  
 Duration(seconds: 5),  
 () => Navigator.*pushReplacement*(  
 context, MaterialPageRoute(builder: (context) => welcome())));  
 } *//<- Creates an object that fetches an image.  
  
 // var image = new Image(  
 // image: AssetImage(  
 // 'asset/quiz.png'),  
 // height:300);* @override  
 Widget build(BuildContext context) {  
 *return* Container(  
 decoration: *const* BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topRight*,  
 end: Alignment.*bottomLeft*,  
 colors: [  
 Colors.*amber*,  
 Colors.*black*,  
 ],  
 )  
 ),  
 child: Column(  
 children: <Widget>[  
 SizedBox(height: 60,),  
 Image(image: AssetImage('images/p2.png'), height: 200),  
 SizedBox(height: 30,),  
 Text("Shiekh Amish Sabir\_090", style: GoogleFonts.*nunito*(fontSize: 20,color: Colors.*white*)),  
  
 SizedBox(  
 height: 10,  
 ),  
 CircularProgressIndicator()  
 ],  
 ));  
 }  
}  
  
*class* welcome *extends* StatelessWidget {  
  
  
 @override  
 Widget build(BuildContext context) {  
 *return* Scaffold(  
 appBar: AppBar(title: Text("QuizApp"),  
 backgroundColor: Colors.*yellow*.shade700,),  
  
backgroundColor: Colors.*amber*,  
 body: welcomepage(),  
 drawer: Drawer(  
 *// Add a ListView to the drawer. This ensures the user can scroll  
 // through the options in the drawer if there isn't enough vertical  
 // space to fit everything.* child: ListView(  
  
 *// Important: Remove any padding from the ListView.* padding: EdgeInsets.*zero*,  
 children: [  
 *new* UserAccountsDrawerHeader(  
 decoration: BoxDecoration(  
 color: Colors.*amber*,  
  
 ),  
 accountName: *new* Text('Shiekh Amish'),  
 accountEmail: *new* Text('amishsabir2@gmail.com'),  
 currentAccountPicture: *new* CircleAvatar(  
 backgroundColor: Colors.*amber*,  
 child: Image.asset('images/profile.png'),  
  
 ),  
  
 ),  
  
 ListTile(  
 title: *const* Text('Home'),  
 onTap: (  
  
 ) {  
 *// Update the state of the app  
 // ...  
 // Then close the drawer* Navigator.*pop*(context);  
 },  
 ),  
 ListTile(  
 title: *const* Text('Source Code'),  
 onTap: () {  
 *// Update the state of the app  
 // ...  
 // Then close the drawer* \_launchURL();  
 },  
 ),  
 ListTile(  
 title: *const* Text('Share'),  
 onTap: (  
  
 ) {  
 *// Update the state of the app  
 // ...  
 // Then close the drawer* \_shareContent();  
 },  
 ),  
  
 ListTile(  
 title: *const* Text('Contact Us'),  
 onTap: () {  
 *// Update the state of the app  
 // ...  
 // Then close the drawer* showAlertDialog(context);  
  
 *// mailto:<email address>?subject=<subject>&body=<body>* },  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}  
\_launchURL() *async* {  
 *const* url = 'https://github.com/ShiekhAmish/FLutter\_Public-/tree/main/Final%20Quiz%20App';  
 *if* (*await* canLaunch(url)) {  
 *await* launch(url);  
 } *else* {  
 *throw* 'Could not launch $url';  
 }  
}  
showAlertDialog(BuildContext context) {  
 *// Create button* Widget okButton = FlatButton(  
 child: Text("OK"),  
 onPressed: () {  
 Navigator.*of*(context).pop();  
 },  
 );  
  
 *// Create AlertDialog* AlertDialog alert = AlertDialog(  
 title: Text("Contact US"),  
 content: Text("Mail To: amishsabir2@gmail.com \n\nTel #: 03096402445"),  
 actions: [  
 okButton,  
 ],  
 );  
  
 *// show the dialog* showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* alert;  
 },  
 );}  
 Route \_createRoute() {  
 *return* PageRouteBuilder(  
 pageBuilder: (context, animation, secondaryAnimation) => QuizApp(),  
 transitionsBuilder: (context, animation, secondaryAnimation, child) {  
 *const* begin = Offset(0.0, 1.0);  
 *const* end = Offset.*zero*;  
 *const* curve = Curves.*ease*;  
  
 *var* tween = Tween(begin: begin, end: end).chain(CurveTween(curve: curve));  
  
 *return* SlideTransition(  
 position: animation.drive(tween),  
 child: child,  
 );  
 },  
 );  
}  
  
*class* welcomepage *extends* StatefulWidget {  
  
 @override  
 \_welcomepageState createState() => \_welcomepageState();  
}  
*final* String \_content =  
 'Made by Shiekh Amish \n\n Get Help From: https://www.youtube.com/c/0673007626pakacademy79 \n\nMail: amishsabir2@gmail.com \n\n Apk File: Hehe';  
  
*void* \_shareContent() {  
 Share.*share*(\_content);  
}  
*class* \_welcomepageState *extends* State<welcomepage> {  
  
  
 @override  
 Widget build(BuildContext context) {  
 *return* Container(  
 decoration: BoxDecoration(  
 image: DecorationImage(  
 image: AssetImage("images/back2.png"),  
 fit: BoxFit.cover,  
 ),  
 ),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 SizedBox(height: 20,),  
 Row(  
  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Container(  
 height: 200,  
 width: 200,  
 decoration: BoxDecoration(  
 image: DecorationImage(  
 image: AssetImage('images/quiz.gif'),  
  
 ),  
 ),  
  
  
 ),  
  
 ],  
 ),  
 SizedBox(height: 120,),  
  
 Column(  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Container(  
 child: TextButton(  
 child: Image.asset("images/play.png"),  
 onPressed: () {  
  
 Navigator.*of*(context).push(\_createRoute()  
 );  
 },  
 )  
  
  
 ),  
  
 ],  
  
 ),  
  
  
  
 Row(  
  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Container(  
 child: TextButton(  
 child: Image.asset("images/share.png"),  
 onPressed: \_shareContent,  
 )  
 ),  
 ],  
 ),  
  
 ],  
 ),  
 ],  
 ),  
 );  
  
 }  
}

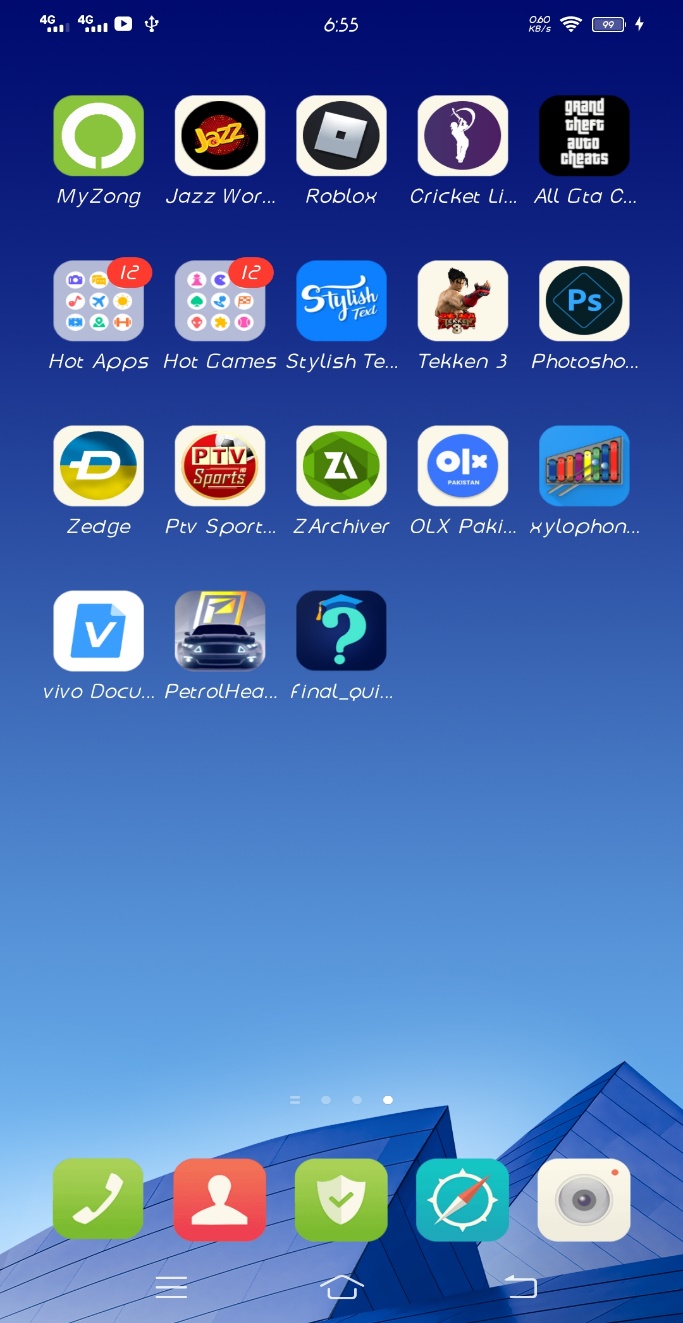
**main.dart**

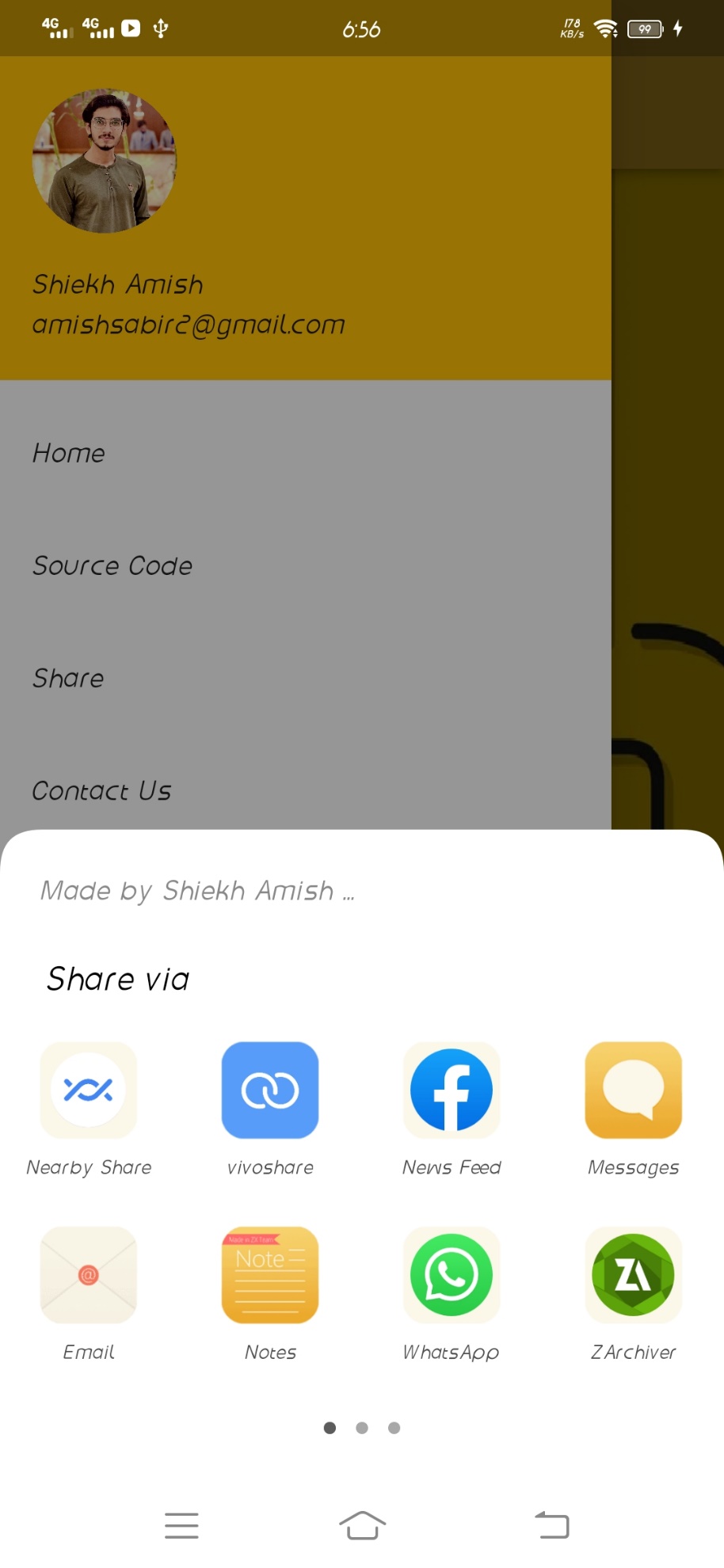
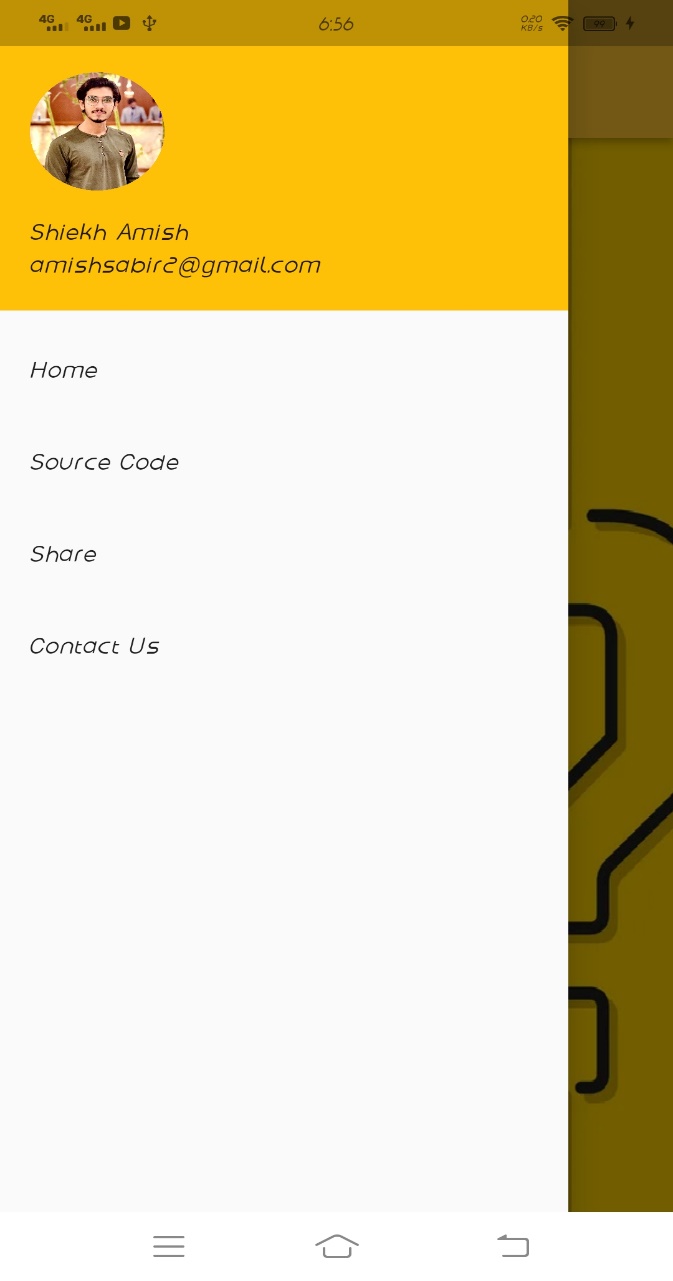
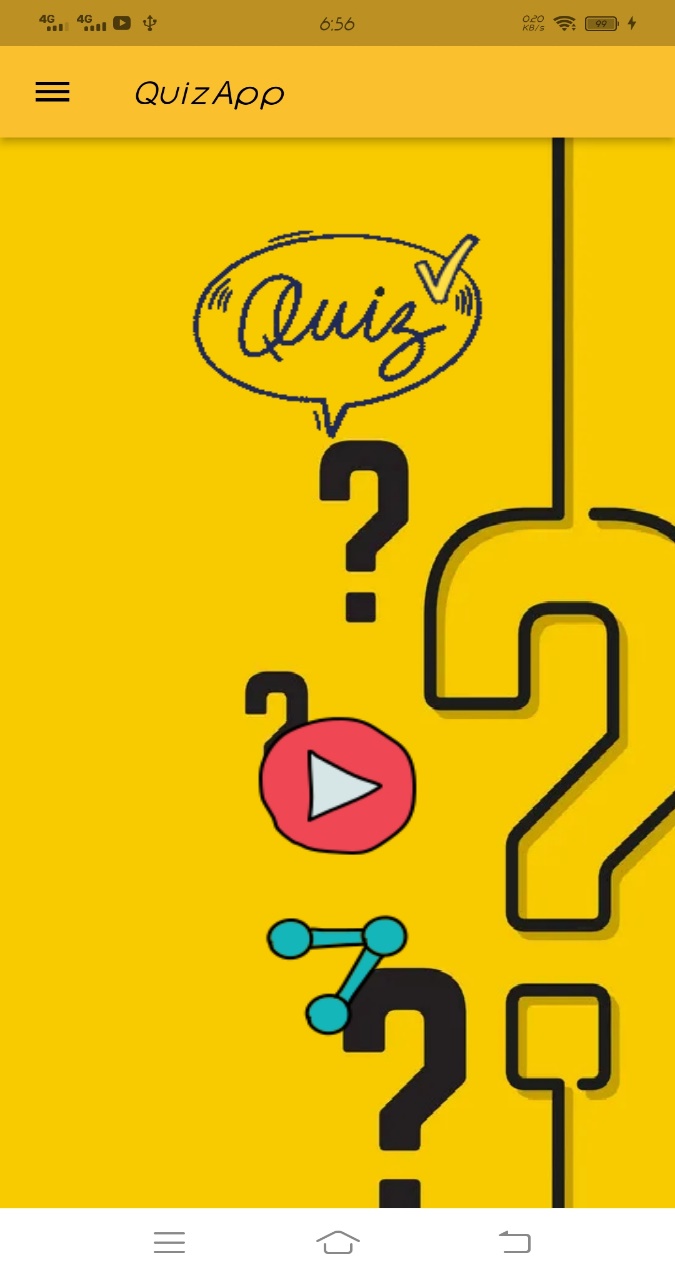
*import* 'dart:async';  
*import* 'package:final\_quiz\_app/welcome.dart';  
*import* 'package:flutter/material.dart';  
*import* 'package:final\_quiz\_app/retry.dart';  
*import* 'quiz\_brain.dart';  
*import* 'dart:ui';  
*import* 'package:flutter\_animated\_button/flutter\_animated\_button.dart';  
*import* 'package:google\_fonts/google\_fonts.dart';  
*import* 'package:simple\_animations/simple\_animations.dart';  
*import* 'package:animated\_text\_kit/animated\_text\_kit.dart';  
*void* main() => runApp(SplashScreen());  
  
*class* QuizApp *extends* StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 *return* MaterialApp(  
 home: Scaffold(  
 backgroundColor: Colors.*amber*,  
 body: SafeArea(child: QuizPage()  
  
 *// Padding(  
 // // padding: EdgeInsets.symmetric(horizontal: 10.0),  
 // child: QuizPage(),  
 // ),* ),  
 ),  
 debugShowCheckedModeBanner: *false*,  
 );  
 }  
}  
  
*class* QuizPage *extends* StatefulWidget {  
 @override  
 \_QuizPageState createState() => \_QuizPageState();  
}  
  
*class* \_QuizPageState *extends* State<QuizPage> {  
 *// DateTime alert;  
  
 // @override  
 // void initState() {  
 // super.initState();  
 // alert = DateTime.now().add(Duration(seconds: 10));  
 // }* QuizBrain quizBrain = QuizBrain();  
 String timer = '';  
 List<Widget> scoreKeeper = [];  
 int correct = 0;  
 int wrong = 0;  
 *// bool change=false;  
 // int next\_que=-1;  
 // int working=0;* List<int> wrong\_answers = [];  
 List<int> wrong\_answers2 = [];  
 List<int> wrong\_selected\_answers = [];  
 List<int> correct\_answers = [];  
 List<int> selected\_answers = [];  
 int count = 0;  
 List<Widget> \_getList(BuildContext context) {  
 List<String> options = quizBrain.getoptions();  
 *// print(options.length);* List<Widget> temp = [];  
 int \_value = -1;  
 *for* (*var* q = 1; q <= options.length; q++) {  
 temp.add(ListTile(  
 title: Text(  
 options[q - 1],  
 style: TextStyle(fontSize: 22, color: Colors.*white*),  
 ),  
 leading: Transform.scale(  
 scale: 2.0,  
 child: Radio(  
 hoverColor: Colors.*amberAccent*,  
 value: q,  
 groupValue: \_value,  
 onChanged: (int value) {  
 checkAnswer(q, context);  
 setState(() {  
 \_value = value;  
 selected\_answers.add(value);  
 *// print()* count += 1;  
 });  
 },  
 ),  
 ))  
 *// new Container(  
 // padding: EdgeInsets.symmetric(horizontal: 25),  
 // decoration: BoxDecoration(  
 // // shape: BoxShape.circle,  
 // borderRadius: BorderRadius.circular(80.0),  
 // ),  
 // width: double.infinity,  
 // height: 80,  
 // child:FlatButton(  
 // color: Color(0xFF00E676),  
 // textColor: Colors.white,  
 // child: new Text(options[q-1],style: TextStyle(fontSize: 25),),  
 // onPressed: () {  
 // checkAnswer(q,context);  
 // },  
 // )  
 // )* );  
 }  
 *return* temp;  
 }  
  
 buttonFunction(id) *async* {  
 *var* d = *await* Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) =>  
 QuizApp2(id, quizBrain, selected\_answers[id])));  
 *// print(d);  
 if* (d[0]) {  
 setState(() {  
 scoreKeeper[d[1]] = Icon(  
 Icons.*check*,  
 color: Colors.*green*,  
 );  
 });  
 correct += 1;  
 wrong -= 1;  
 print(wrong\_answers);  
 correct\_answers.add(selected\_answers[id]);  
 int index = wrong\_answers.indexWhere(  
 (i) => i.toString().contains(wrong\_answers2[id].toString()));  
 *if* (index > -1) {  
 wrong\_answers.removeAt(index);  
 }  
  
 *// wrong\_answers[]* }  
 }  
  
 *void* checkAnswer(int userPickedAnswer, BuildContext context) {  
 int correctAnswer = quizBrain.getCorrectAnswer();  
 setState(() {  
 *if* (quizBrain.isFinished() == *true*) {  
 *// Alert.alert(context, title: "Hello", content: "this is a alert")  
 // .then((\_) => Alert.toast(context, "You just click ok"));  
 // Alert(  
 // context: this.context,  
 // title: 'Finshed',  
 // desc: 'You\'ve reached the end of the quiz.\nTrue Answer: $correct \n Wrong Answer: $wrong',  
 // ).show();  
 if* (userPickedAnswer == correctAnswer) {  
 correct\_answers.add(quizBrain.get\_number());  
 correct += 1;  
 scoreKeeper.add(Icon(  
 Icons.*check*,  
 color: Colors.*green*,  
 ));  
 } *else* {  
 wrong\_answers.add(quizBrain.get\_number());  
 wrong\_selected\_answers.add(userPickedAnswer);  
 wrong += 1;  
 scoreKeeper.add(IconButton(  
 icon: Icon(  
 Icons.*close*,  
 color: Colors.*red*,  
 ),  
 onPressed: () {  
 *// var data = Navigator.push(  
 // context,  
 // MaterialPageRoute(builder: (context) => QuizApp2(count,quizBrain,selected\_answers[count])));  
  
 // print("Wrong");  
 // setState(() {  
 // quizBrain.change\_question(quizBrain.get\_number());  
 // print(quizBrain.get\_number());  
 // \_volume += 10;  
 // });* },  
 ));  
 }  
 *// print(wrong);  
 // print(correct);  
 // quizBrain.reset();  
 // quizBrain.shuffle();  
 // scoreKeeper = [];  
 // showAlert(context);* timer = 'complete';  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => SecondRoute(quizBrain, correct, wrong,  
 correct\_answers, wrong\_answers, timer)));  
 } *else* {  
 wrong\_answers2.add(quizBrain.get\_number());  
 *if* (userPickedAnswer == correctAnswer) {  
 correct\_answers.add(quizBrain.get\_number());  
 correct += 1;  
 scoreKeeper.add(Icon(  
 Icons.*check*,  
 color: Colors.*green*,  
 ));  
 } *else* {  
 wrong\_answers.add(quizBrain.get\_number());  
  
 scoreKeeper.add(  
 *// score\_wrong(count)  
 new* FAB(  
 id: count,  
 onPressed: buttonFunction,  
 ));  
 wrong += 1;  
 }  
 quizBrain.nextQuestion();  
 }  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 *// quizBrain.shuffle();  
 return* Container(  
 decoration: BoxDecoration(  
 image: DecorationImage(  
 image: AssetImage("images/back3.png"),  
 fit: BoxFit.cover,  
 ),  
 ),  
 child:Column(  
  
 mainAxisAlignment: MainAxisAlignment.spaceAround,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: <Widget>[  
 TweenAnimationBuilder<Duration>(  
 duration: Duration(minutes: 1),  
 tween: Tween(begin: Duration(minutes: 1), end: Duration.*zero*),  
 onEnd: () {  
 *// print('Timer ended');  
 // quizBrain.reset();  
 // quizBrain.shuffle();  
 // scoreKeeper = [];* timer = 'time';  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => SecondRoute(quizBrain, correct,  
 wrong, correct\_answers, wrong\_answers, timer)));  
 },  
 builder: (BuildContext context, Duration value, Widget child) {  
 *var* minutes = value.inMinutes;  
 *var* seconds = value.inSeconds % 60;  
 *return* Padding(  
 padding: *const* EdgeInsets.symmetric(vertical: 5),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Icon(  
 Icons.*alarm*,  
 color: Colors.*red*,  
 size: 50,  
 ),  
 Text('$minutes:$seconds',  
 textAlign: TextAlign.center,  
 style: TextStyle(  
 color: Colors.*white*,  
 fontWeight: FontWeight.*bold*,  
 fontSize: 50))  
 ],  
 ));  
 }),  
 Container(  
 height: 50,  
 width: 150,  
 child: AnimatedButton(  
 height: 70,  
 width: 200,  
 text: 'Stop',  
 isReverse: *true*,  
 selectedTextColor: Colors.*black*,  
 transitionType: TransitionType.LEFT\_TOP\_ROUNDER,  
 textStyle: GoogleFonts.*nunito*(  
 fontSize: 28,  
 letterSpacing: 5,  
 color: Colors.*white*,  
 fontWeight: FontWeight.*w300*),  
 onPress: () {  
  
 timer = 'stop';  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => SecondRoute(quizBrain, correct,  
 wrong, correct\_answers, wrong\_answers, timer)));  
 },  
 ),  
  
  
 ),  
  
 *// RaisedButton(  
 // onPressed: () {},  
 // textColor: Colors.white,  
 // padding: const EdgeInsets.all(0.0),  
 // child: Container(  
 //  
 // decoration: const BoxDecoration(  
 // gradient: LinearGradient(  
 // colors: <Color>[  
 // Color(0xFF0D47A1),  
 // Color(0xFF1976D2),  
 // Color(0xFF42A5F5),  
 // ],  
 // ),  
 // ),  
 // padding: const EdgeInsets.all(10.0),  
 // child:  
 // const Text('Stop Quiz', style: TextStyle(fontSize: 20)),  
 // ),  
 // ),* Expanded(  
 flex: 1,  
 child: Padding(  
 padding: EdgeInsets.all(10.0),  
 child: Center(  
 child: PlayAnimation<double>(  
 tween: Tween(begin: 0.0, end: 200.0),  
 duration: *const* Duration(seconds: 4),  
 curve: Curves.*easeOut*,  
 builder: (context, child, value)  
 {  
 *return* Container(  
  
 color: Colors.*amber*.withOpacity(0.8),  
 height: 100,  
 width: 550,  
 child: Center(  
 child: Text(  
  
 quizBrain.getQuestionText(),  
 textAlign: TextAlign.center,  
 style: GoogleFonts.*nunito*(  
 fontSize: 25.0,  
 color: Colors.*black*,  
  
 ),  
  
 ),  
 ),  
  
 );  
 }  
 ),  
 ),  
 ),  
 ),  
  
 Column(  
 children: \_getList(context),  
 ),  
  
 Row(  
 children: scoreKeeper,  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
*// /\*  
// question1: 'You can lead a cow down stairs but not up stairs.', false,  
// question2: 'Approximately one quarter of human bones are in the feet.', true,  
// question3: 'A slug\'s blood is green.', true,  
// \*/  
class* SecondRoute *extends* StatelessWidget {  
 QuizBrain quizBrain;  
 int correct;  
 int wrong;  
 String timer;  
 List<int> wrong\_answers;  
 List<int> correct\_answers;  
  
 *// final List<int> selected\_answer;* SecondRoute(*this*.quizBrain, *this*.correct, *this*.wrong, *this*.correct\_answers,  
 *this*.wrong\_answers, *this*.timer,  
 {Key key})  
 : *super*(key: key);  
 Widget check\_timer(bool check) {  
 *if* (timer == 'time' && check == *true*) {  
 *return* Container(  
 child: Text(  
 'Time Over',  
 style: GoogleFonts.*nunito*(  
 fontSize: 30, color: Colors.*red*, fontWeight: FontWeight.*bold*),  
 ),  
 );  
 } *else if* (timer == 'stop' && check == *true*) {  
 *return* Container(  
 child: Text(  
 'Quiz Stopped',  
 style: GoogleFonts.*nunito*(  
 fontSize: 30, color: Colors.*red*, fontWeight: FontWeight.*bold*),  
 ),  
 );  
 } *else if* (timer == 'complete' && check == *true*) {  
 *return* Container(  
 child: Center(  
 child: TextLiquidFill(  
 text: 'Quiz Complete',  
 waveDuration: Duration(seconds: 5),  
 waveColor: Colors.*amber*,  
 boxBackgroundColor: Colors.*black*,  
 textStyle: TextStyle(  
 fontSize: 50.0,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 ),  
 *// child: Text(  
 // 'Quiz Completed',  
 // style: GoogleFonts.nunito(  
 // fontSize: 30, color: Colors.white, fontWeight: FontWeight.bold),  
 // ),* );  
 } *else if* (!check && timer != 'complete') {  
 *return* Container(  
 child: Column(  
 children: <Widget>[  
 Text(  
 '${10 - (correct + wrong)}',  
 style: GoogleFonts.*nunito*(fontSize: 30, color: Colors.*white*),  
 ),  
 Text(  
 'Unattempt Question',  
 style: GoogleFonts.*nunito*(  
 fontSize: 30, color: Colors.*white*, fontWeight: FontWeight.*bold*),  
 )  
 ],  
 ));  
 } *else* {  
 *return* SizedBox(  
 height: 0,  
 );  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 *return* Scaffold(  
 backgroundColor: Colors.*amber*,  
 appBar: AppBar(  
 title: Text("Results"),  
 backgroundColor: Colors.*yellow*.shade700,  
 leading: *new* IconButton(  
 icon: *new* Icon(Icons.*arrow\_back*),  
 onPressed: () {  
 Navigator.*push*(  
 context, MaterialPageRoute(builder: (context) => welcome()));  
 }),  
 ),  
 body: Center(  
  
 child: SingleChildScrollView(  
 child: Column(  
 children: <Widget>[  
 Container(  
 color: Colors.*black*.withOpacity(0.5),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: <Widget>[  
 check\_timer(*true*),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: <Widget>[  
 Container(  
 child: Text(  
 '${correct}',  
 style: TextStyle(fontSize: 30, color: Colors.*white*),  
 ),  
 ),  
 Container(  
 child: Text('${wrong}',  
 style: TextStyle(fontSize: 30, color: Colors.*white*)),  
 ),  
 ],  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: <Widget>[  
 Container(  
 child: Text("True Answers",  
 style: TextStyle(  
 fontSize: 20,  
 color: Colors.*white*,  
 fontWeight: FontWeight.*bold*)),  
 ),  
 Container(  
 child: Text("Wrong Answers",  
 style: TextStyle(  
 fontSize: 20,  
 color: Colors.*white*,  
 fontWeight: FontWeight.*bold*)),  
 ),  
 ],  
 ),  
 check\_timer(*false*),  
 *// Container(  
 // child:Text("True Answers",style: TextStyle(fontSize: 20,color: Colors.white)),  
 // ),  
 // Container(  
 // child:Text("Unattempt Questions",style: TextStyle(fontSize: 20,color: Colors.white)),  
 // ),* ],  
 ),  
 ),  
 SizedBox(  
 height: 20,  
 ),  
 *// SingleChildScrollView(* Column(  
 children: quizBrain.getallquestion(wrong\_answers, 0),  
 ),  
 *// ),  
 // SingleChildScrollView(* Column(  
 children: quizBrain.getallquestion(correct\_answers, 1),  
 )  
 *// )* ],  
 ))),  
 );  
 }  
}  
*// class score\_wrong extends StatefulWidget {  
// int counter;//if you have multiple values add here  
// // final List<int> selected\_answer;  
// score\_wrong(this.counter, {Key key}): super(key: key);  
// @override  
// \_PlayButtonState createState() => \_PlayButtonState();  
// }  
// class \_PlayButtonState extends State<score\_wrong> {  
// IconData icon;  
//  
// @override  
// void initState() {  
// icon = Icons.close;  
// super.initState();  
// }  
//  
// @override  
// Widget build(BuildContext context) {  
// return Center(  
// child: FlatButton(  
// child: Icon(icon,color: Colors.red,),  
// onPressed: () {  
// print(quizBrain.getQuestionText\_index(widget.counter));  
// Navigator.push(  
// context,  
// MaterialPageRoute(builder: (context) => QuizApp2()));  
// },  
// ),  
// );  
// }  
// }  
  
class* FAB *extends* StatelessWidget {  
 *final* int id;  
 *final* Function(int) onPressed;  
 *// final String buttonText;  
  
 const* FAB({*this*.id, *this*.onPressed});  
  
 @override  
 Widget build(BuildContext context) {  
 *return* SizedBox(  
 width: 30,  
 child: IconButton(  
 icon: Icon(  
 Icons.*close*,  
 color: Colors.*red*,  
 ),  
 onPressed: () {  
 onPressed(*this*.id);  
 },  
 ),  
 );  
 *// IconButton(  
 //  
 // icon: Icon(  
 // Icons.close,  
 // color: Colors.red,  
 //  
 // ),  
 // onPressed: () {onPressed(this.id);},  
 // );* }  
}

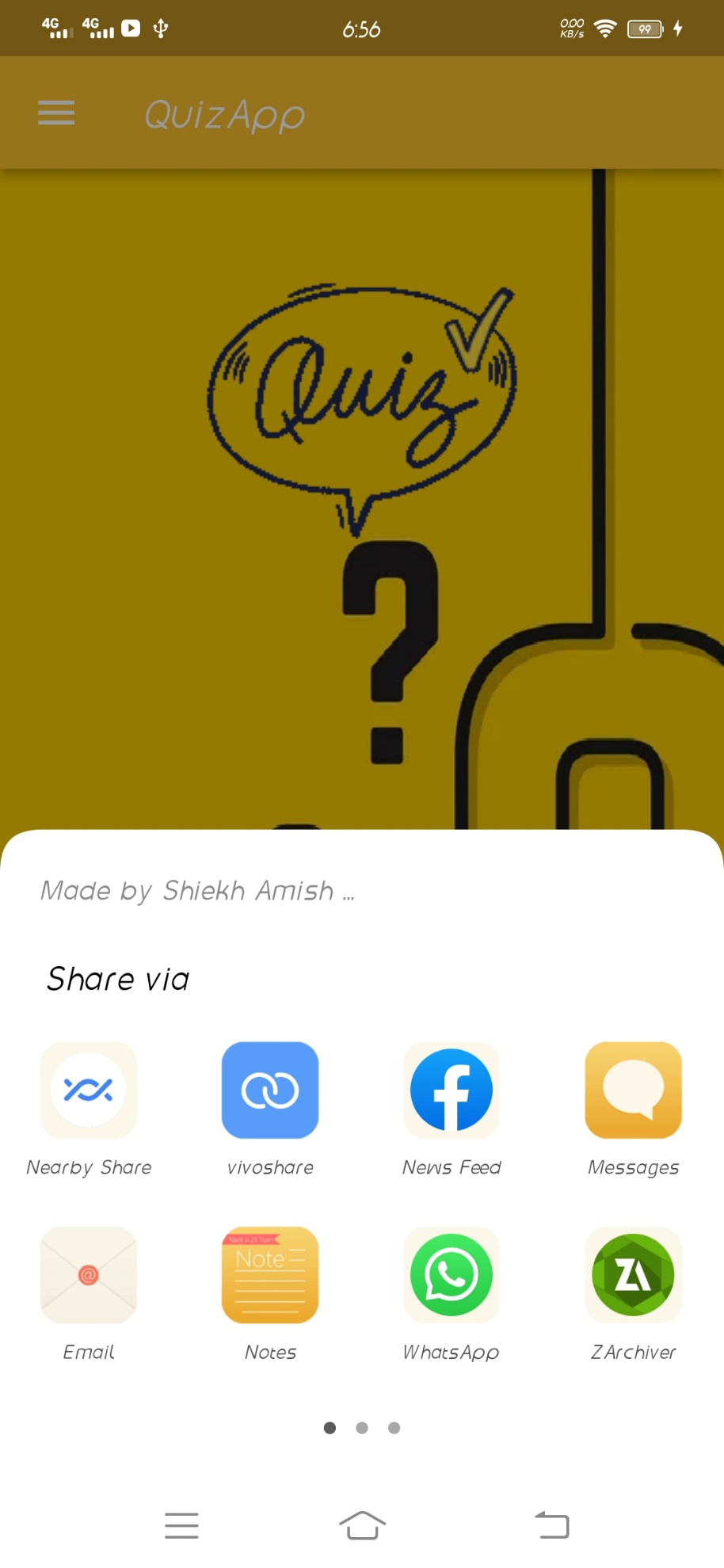
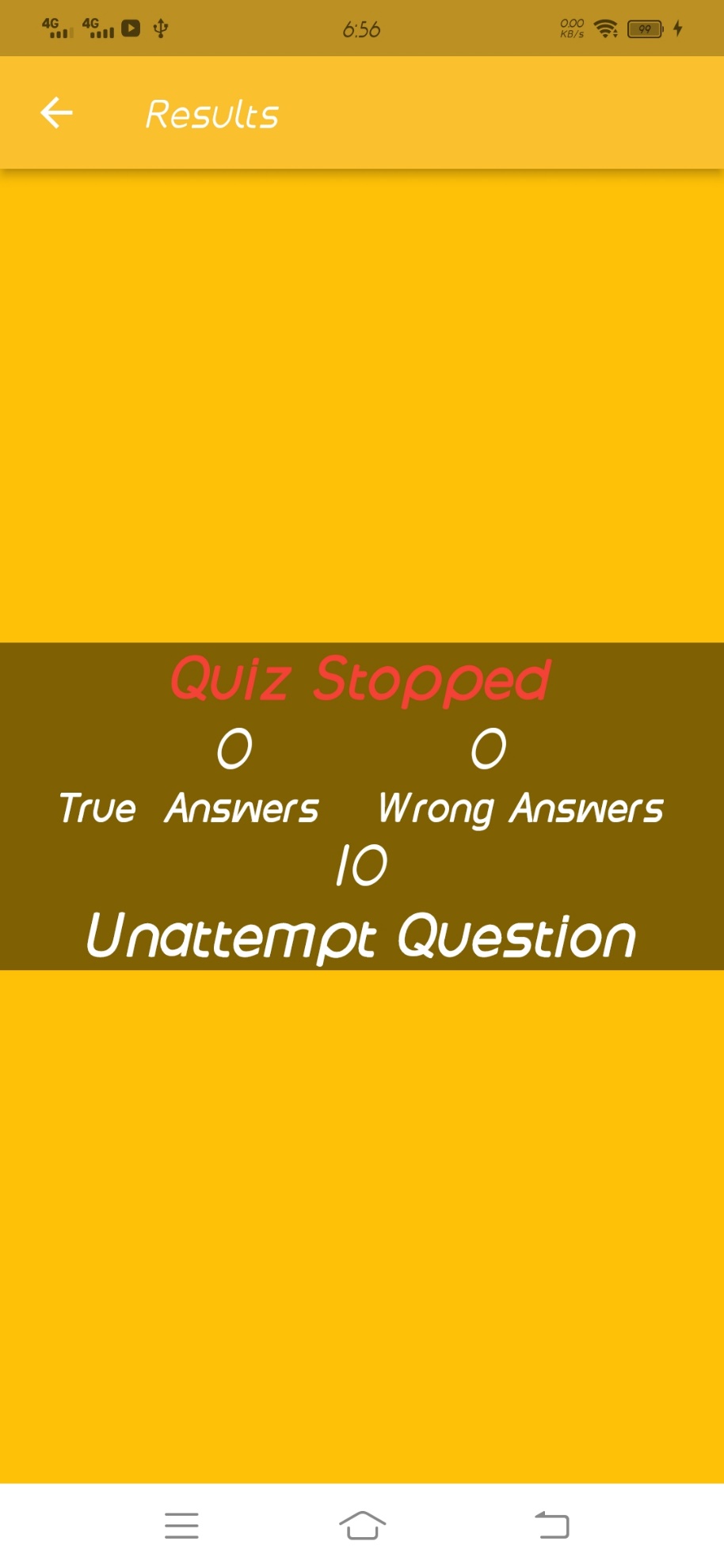
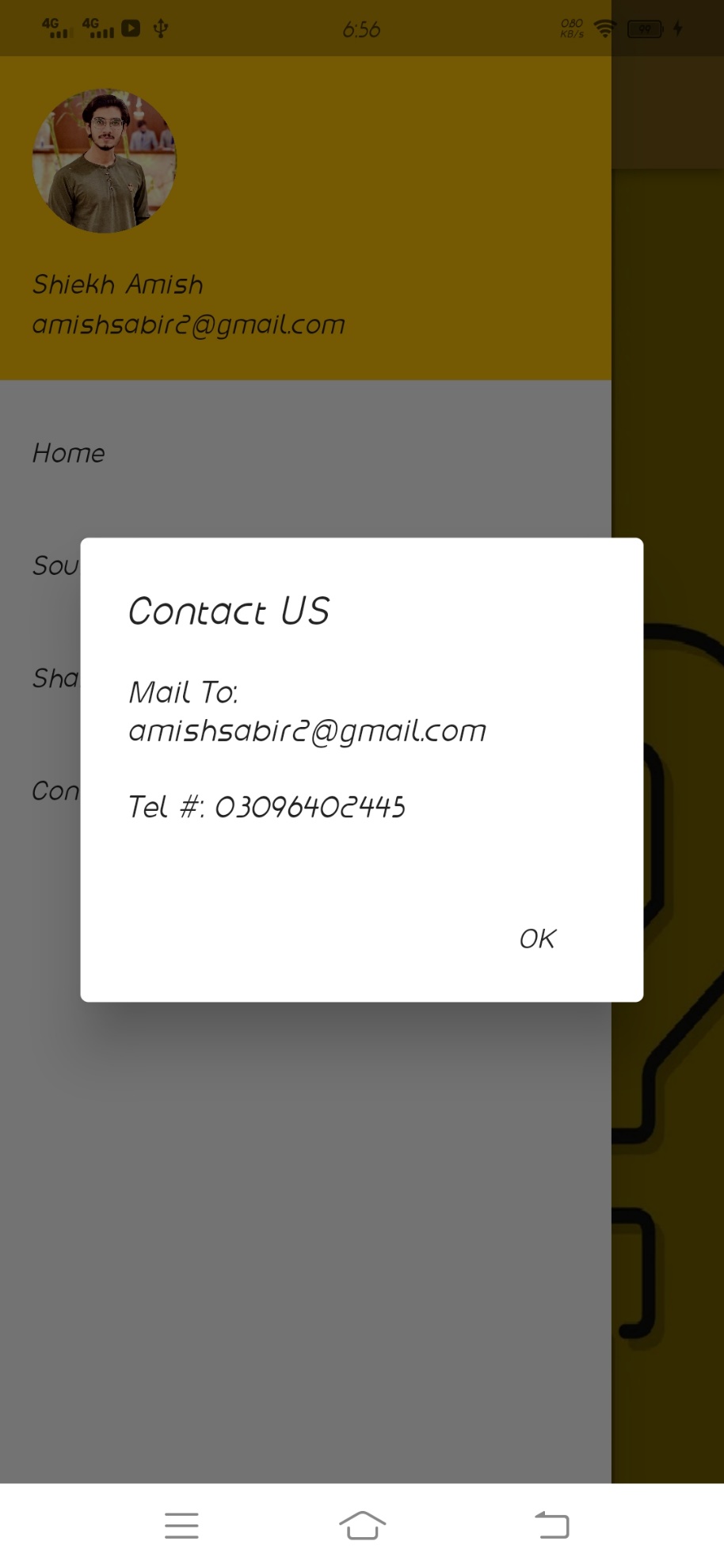
**Quiz Brain.dart**

*import* 'package:flutter/cupertino.dart';  
*import* 'package:final\_quiz\_app/question.dart';  
*import* 'package:flutter/material.dart';  
*import* 'dart:math';  
  
*class* QuizBrain {  
 int \_questionNumber = 0;  
  
 List<Question> \_questionBank = [  
 Question('Code written on Flutter requires Javascript Interpreter',  
 ["True", "False"], 2),  
 Question("Flutter doesn't support 60fps animations",  
 ["True", "False"], 2),  
 Question('Flutter is an app development library',  
 ["True", "False"], 2),  
 Question('Flutter is developed by Facebook',  
 ["True", "False"], 2),  
 Question('Flutter uses one code base', ["True", "False"], 1),  
 Question('Flutter’s engine, written primarily in C++, provides low-level rendering support using Google’s Skia graphics library.',  
 ["True", "False"], 1),  
 Question('A notable feature of the Dart platform is its support for hot reload',  
 ["True", "False"], 1),  
 Question(  
 'Flutter supports cross platform development',  
 ["True", "False"],  
 1),  
 Question('Flutter is only supported on Windows.',  
 ["True", "False"], 2),  
 Question('Flutter is Free',  
 ["True", "False"], 1)  
 *// Question('You can lead a cow down stairs but not up stairs.', false),  
 // Question('Approximately one quarter of human bones are in the feet.', true),  
 // Question('A slug\'s blood is green.', true),  
 // Question('Buzz Aldrin\'s mother\'s maiden name was \"Moon\".', true),  
 // Question('It is illegal to pee in the Ocean in Portugal.', true),  
 // Question(  
 // 'No piece of square dry paper can be folded in half more than 7 times.',  
 // false),  
 // Question(  
 // 'In London, UK, if you happen to die in the House of Parliament, you are technically entitled to a state funeral, because the building is considered too sacred a place.',  
 // true),  
 // Question(  
 // 'The loudest sound produced by any animal is 188 decibels. That animal is the African Elephant.',  
 // false),  
 // Question(  
 // 'The total surface area of two human lungs is approximately 70 square metres.',  
 // true),  
 // Question('Google was originally called \"Backrub\".', true),  
 // Question(  
 // 'Chocolate affects a dog\'s heart and nervous system; a few ounces are enough to kill a small dog.',  
 // true),  
 // Question(  
 // 'In West Virginia, USA, if you accidentally hit an animal with your car, you are free to take it home to eat.',  
 // true),* ]..shuffle();  
  
 *void* nextQuestion() {  
 *if* (\_questionNumber < \_questionBank.length - 1) {  
 \_questionNumber++;  
 }  
 }  
 *// void initialize\_number()* String getQuestionText() {  
 *return* \_questionBank[\_questionNumber].questionText;  
 }  
  
 String getQuestionText\_index(int index) {  
 *return* \_questionBank[index].questionText;  
 }  
  
 int get\_number() {  
 *return* \_questionNumber;  
 }  
  
 int getCorrectAnswer() {  
 *return* \_questionBank[\_questionNumber].index;  
 }  
  
 int getCorrectAnswer\_index(int index) {  
 *return* \_questionBank[index].index;  
 }  
  
 List<String> getoptions() {  
 *return* \_questionBank[\_questionNumber].data;  
 }  
  
 List<String> getoptions\_index(int index) {  
 *return* \_questionBank[index].data;  
 }  
  
 bool isFinished() {  
 *if* (\_questionNumber >= \_questionBank.length - 1) {  
 *return true*;  
 } *else* {  
 *return false*;  
 }  
 }  
  
 List<Widget> getallquestion(List<int> wrongans, int type) {  
 List<Widget> temp = [];  
 *for* (int i = 0; i < wrongans.length; i++) {  
 int t = wrongans[i];  
 *if* (t != -10) {  
 *if* (i == 0 && type == 0) {  
 temp.add(Text(  
 "Wrong Answers",  
 style: TextStyle(  
 fontSize: 30, color: Colors.*black*, fontWeight: FontWeight.*bold*),  
 ));  
 } *else if* (i == 0 && type == 1) {  
 temp.add(Text(  
 "Correct Answers",  
 style: TextStyle(  
 fontSize: 30, color: Colors.*black*, fontWeight: FontWeight.*bold*),  
 ));  
 }  
 temp.add(Container(  
 color: Colors.*amberAccent*,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: <Widget>[  
 Text(  
 \_questionBank[t].questionText,  
 style:  
 TextStyle(fontSize: 20, fontWeight: FontWeight.*bold*),  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 "Answer: ",  
 style: TextStyle(  
 fontSize: 20, fontWeight: FontWeight.*bold*),  
 ),  
 Text(  
 \_questionBank[t].data[\_questionBank[t].index - 1],  
 style: TextStyle(fontSize: 20),  
 ),  
 ],  
 ),  
 SizedBox(  
 height: 50,  
 child: Divider(  
 color: Colors.*black*,  
 height: 2,  
 ),  
 )  
 ],  
 ))  
 *// new Text(+": "+\_questionBank[t].data[\_questionBank[t].index-1],style: TextStyle(fontSize: 20,backgroundColor: Colors.black38,color: Colors.white),)* );  
 }  
 }  
 *return* temp;  
 }  
  
 *void* reset() {  
 \_questionNumber = 0;  
 }  
}  
  
*// void nextQuestion() {  
// if (\_questionNumber < \_questionBank.length - 1) {  
// \_questionNumber++;  
// }  
// }  
//  
// String getQuestionText() {  
// return \_questionBank[\_questionNumber].questionText;  
// }  
//  
// bool getCorrectAnswer() {  
// return \_questionBank[\_questionNumber].questionAnswer;  
// }  
  
//TODO: Step 3 Part A - Create a method called isFinished() here that checks to see if we have reached the last question. It should return (have an output) true if we've reached the last question and it should return false if we're not there yet.  
  
//TODO: Step 3 Part B - Use a print statement to check that isFinished is returning true when you are indeed at the end of the quiz and when a restart should happen.  
  
//TODO: Step 4 Part B - Create a reset() method here that sets the questionNumber back to 0.  
//}*

**Screenshots**

****



****