**Xây dựng ứng dụng làm trắc nghiệm**

**Phát thảo ý tưởng**

A screenshot of a quiz

Description automatically generated

Chúng ta sẽ làm chức năng chính cần thiết đầu tiên, layout dựa vào wireframe ở trên.

**Tạo đối tượng**

Chúng ta tạo class ***Question.dart*** với các trường thông tin cần sử dụng trong đó:

* question là câu hỏi
* options là danh sách đáp án
* answer là đáp án đúng
* feedback là lời giải

class Question {

int id;

String question;

List<String> options;

int answer;

String feedback;

Question(

{required this.id,

required this.question,

required this.options,

required this.answer,

required this.feedback});

}

Đồng thời mình sẽ tạo data câu hỏi

final List<Question> questionsData = [

Question(

id: 1,

question:

"The three pillars of empiricism are: The three pillars of empiricism are: The three pillars of empiricism are: The three pillars of empiricism are:",

options: [

'Planning, Inspection, Adaptation.',

'Transparency, Eliminating Waste, Kaizen.',

'Inspection, Transparency, Adaptation.',

'Planning, Demonstration, Retrospective.',

'Respect For People, Kaizen, Eliminating Waste.'

],

answer: 1,

feedback:

"Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known.\nThree pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation."),

Question(

id: 2,

question: "Who has the final say on the order of the Product Backlog?",

options: [

'The Scrum Master.',

'The Product Owner.',

'The Stakeholders.',

'The Developers.',

'The CEO.'

],

answer: 1,

feedback:

"The Product Owner is the sole person responsible for ordering the Product Backlog."),

Question(

id: 3,

question: "What is the recommended size for a Scrum Team?",

options: [

"Minimum of 7.",

"9.",

"Typically 10 or fewer people.",

"7 plus or minus 2."

],

answer: 2,

feedback:

"A Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. Generally smaller teams communicate better and are more productive")

];

**Màn Home**

Ở main.dart các bạn chỉnh sửa app điều hướng home là HomeScreen

class \_MyAppState extends State<MyApp> {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Quizzy',

theme: ThemeData(

primarySwatch: Colors.deepPurple,

),

home: HomeScreen(),

);

}

}

Mình tạo một StatefulWidget là HomeScreen. Chứa một Button để start bắt đầu làm trắc nghiệm.

class HomeScreen extends StatefulWidget {

const HomeScreen({Key? key}) : super(key: key);

@override

\_HomeScreenState createState() => \_HomeScreenState();

}

class \_HomeScreenState extends State<HomeScreen> {

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: ElevatedButton.icon(

onPressed: () {

Navigator.push(

context, MaterialPageRoute(builder: (\_) => QuizScreen()));

},

label: Text('Start'),

icon: Icon(Icons.arrow\_forward),

),

),

);

}

}

Code main.dart

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatefulWidget {  
 *// This widget is the root of your application.* @override  
 \_MyAppState createState() => \_MyAppState();  
}  
  
class \_MyAppState extends State<MyApp> {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 title: 'Quizzy',  
 theme: ThemeData(  
 primarySwatch: Colors.*deepPurple*,  
 ),  
 home: HomeScreen(),  
 );  
 }  
}  
class HomeScreen extends StatefulWidget {  
 const HomeScreen({Key? key}) : super(key: key);  
  
 @override  
 \_HomeScreenState createState() => \_HomeScreenState();  
}  
  
class \_HomeScreenState extends State<HomeScreen> {  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Center(  
 child: ElevatedButton.icon(  
 onPressed: () {  
 *// Navigator.push(  
 // context, MaterialPageRoute(builder: (\_) => QuizScreen()));* },  
 label: Text('Start'),  
 icon: Icon(Icons.*arrow\_forward*),  
 ),  
 ),  
 );  
 }  
}

A screenshot of a cell phone

Description automatically generated

**Màn làm trắc nghiệm**

Như ý tưởng mình sẽ tạo một appbar

appBar: AppBar(

title: Text("QUIZ"),

elevation: 0,

),

Mình tạo một biến \_currentIndex để lưu lại vị trí câu hỏi hiện tại trong list câu hỏi.

int \_currentIndex = 0;

Mình lấy ra câu hỏi trong danh sách câu hỏi bằng cách

final question = questionsData[\_currentIndex];

Phần UI câu hỏi mình sẽ tạo một Row

Row(

children: [

CircleAvatar(

backgroundColor: Colors.deepPurple,

child: Text("${\_currentIndex + 1}"),

),

SizedBox(width: 16.0),

Expanded(

child: Text(question.question),

)

],

),

Với danh sách câu trả lời, mình sẽ tạo một Card

Card(

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

...question.options.map(

(option) => RadioListTile(

title: Text(option),

value: option,

groupValue: \_answers[\_currentIndex],

onChanged: (value) {

setState(() {

\_answers[\_currentIndex] = option;

});

},

),

)

],

),

),

Để lưu lại kết quả lựa chọn mfinh tạo một biến \_answers

final Map<int, dynamic> \_answers = {};

Nút chuyển câu hỏi:

Expanded(

child: Container(

alignment: Alignment.bottomCenter,

child: ElevatedButton.icon(

onPressed: \_handleNext,

label: Text('Next'),

icon: Icon(Icons.arrow\_forward),

),

),

)

Mình xử lý logic khi bấn nút chuyển câu hỏi như sau:

* Người dùng không chọn câu trả lời nào => Hiện dialog nhắc nhở.
* Còn câu hỏi trong list câu hỏi => Hiện thị câu hỏi kế tiếp
* Đã hết câu hỏi => Hiện thị trang kết quả

void \_handleNext() {

if (\_answers[\_currentIndex] == null) {

\_showAlertDialog();

return;

}

if (\_currentIndex < (questionsData.length - 1)) {

setState(() {

\_currentIndex++;

});

} else {

Navigator.of(context).pushReplacement(MaterialPageRoute(

builder: (context) => QuizResultScreen(),

));

}

}

Khi mình set state cập nhật \_currentIndex, câu hỏi sẽ được update lại.

Dùng [showDialog](https://api.flutter.dev/flutter/material/AlertDialog-class.html) để hiện dialog

void \_showAlertDialog() {

showDialog<String>(

context: context,

builder: (BuildContext context) => AlertDialog(

title: const Text('Warning'),

content: const Text('You must attempt all questions to continue'),

actions: <Widget>[

TextButton(

onPressed: () => Navigator.pop(context, 'OK'),

child: const Text('OK'),

),

],

),

);

}

Và kết quả như sau

A screenshot of a cell phone

Description automatically generated

Như vậy cấu trúc chương trình gồm 3 file:

A screenshot of a computer

Description automatically generated

Code main.dart

A screenshot of a computer program

Description automatically generated

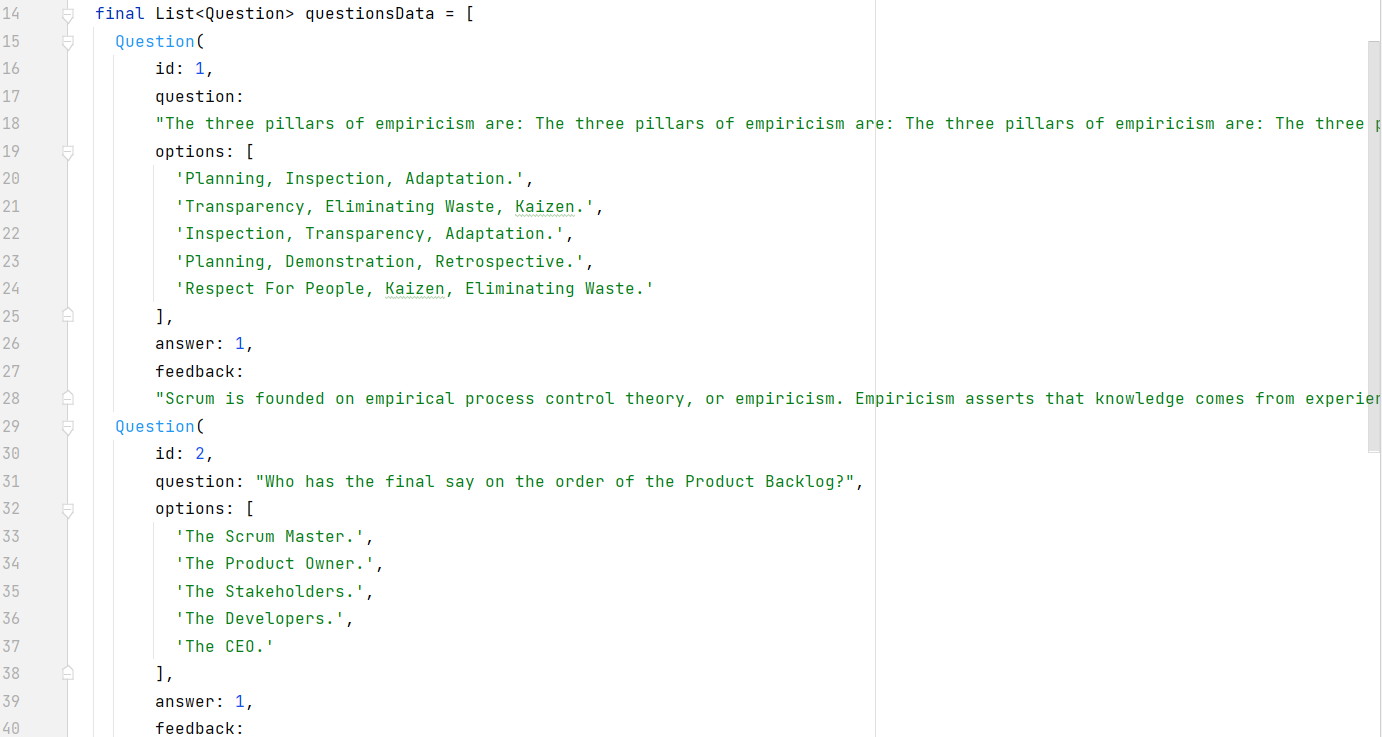
A computer screen shot of a program

Description automatically generated

Code Question như sau:

A screen shot of a computer code

Description automatically generated



Code QuízScreen.dart

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A computer screen shot of a code

Description automatically generated