import 'dart:async';

import 'dart:convert';

import 'dart:core';

import 'dart:io';

import 'package:flutter/material.dart';

import 'package:path\_provider/path\_provider.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

        title: 'Flutter Demo',

        theme: ThemeData(

          primarySwatch: Colors.blue,

        ),

        initialRoute: '/',

        routes: <String, WidgetBuilder>{

          '/': (context) => Dictionary(storage: Storage(), title: 'Dictionary'),

          // '/detail/': (context) => WordDetail(word: ,)

          '/add\_word': (context) => AddWord(storage: Storage())

        });

  }

}

class Storage {

  Future<String> get \_localPath async {

    final directory = await getApplicationDocumentsDirectory();

    print(directory.path);

    return directory.path;

  }

  Future<File> get \_localFile async {

    final path = await \_localPath;

    return File('$path/mydictionary.json');

  }

  Future<File> addWord(Word w) async {

    final file = await \_localFile;

    // Write the file

    return file.writeAsString('$w');

  }

  Future<dynamic> loadMyWords() async {

    try {

      final file = await \_localFile;

      // Read the file

      final contents = await file.readAsString();

      return int.parse(contents);

    } catch (e) {

      // If encountering an error, return 0

      return 0;

    }

  }

  Future<dynamic> loadDefautWords(BuildContext context) async {

    final lst =

        await DefaultAssetBundle.of(context).loadString('db/dictionary.json');

    final \_dictionaryData = json.decode(lst);

    return \_dictionaryData;

  }

}

class Word {

  String key;

  String value;

  Word(this.key, this.value);

}

class WordDetail extends StatelessWidget {

  const WordDetail({Key? key, required this.word}) : super(key: key);

  final Word word;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        appBar: AppBar(

          title: Text("Detail"),

        ),

        body: Container(

            padding: EdgeInsets.only(top: 20, left: 30, right: 30),

            child: Column(

              children: [

                Align(

                  alignment: Alignment.bottomLeft,

                  child: Text(

                    word.key,

                    style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30),

                  ),

                ),

                Divider(),

                Align(alignment: Alignment.bottomLeft, child: Text(word.value))

              ],

            )));

  }

}

class AddWord extends StatefulWidget {

  const AddWord({Key? key, required this.storage}) : super(key: key);

  final Storage storage;

  @override

  State<AddWord> createState() => \_AddWordState();

}

class \_AddWordState extends State<AddWord> {

  TextEditingController \_wordController = new TextEditingController();

  TextEditingController \_meaningController = new TextEditingController();

  Word newWord = new Word("", "");

  Future<File> \_addWord() {

    setState(() {

      newWord.key = \_wordController.text;

      newWord.value = \_meaningController.text;

    });

    // Write the variable as a string to the file.

    return widget.storage.addWord(newWord);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        appBar: AppBar(

          title: Text("Add Word"),

        ),

        body: Container(

            padding: EdgeInsets.only(top: 20, left: 30, right: 30),

            child: ListView(

              children: [

                TextField(

                  controller: \_wordController,

                  decoration: InputDecoration(

                    hintText: "Word",

                    border: OutlineInputBorder(),

                  ),

                ),

                Container(

                  padding: EdgeInsets.only(top: 10, bottom: 10),

                  child: TextField(

                    controller: \_meaningController,

                    maxLines: 15,

                    decoration: InputDecoration(

                      hintText: "Meaning",

                      border: OutlineInputBorder(),

                    ),

                  ),

                ),

                OutlinedButton(

                    onPressed: () {

                      // \_addWord();

                      showDialog(

                          context: context,

                          builder: (BuildContext context) {

                            return AlertDialog(

                              title: Text("Bi qua"),

                              content:

                                  Text("Chua biet lam nhu the nao thay oi"),

                            );

                          });

                    },

                    child: Text("Add Word"))

              ],

            )));

  }

}

class Dictionary extends StatefulWidget {

  const Dictionary({Key? key, required this.storage, required this.title})

      : super(key: key);

  final Storage storage;

  final String title;

  @override

  State<Dictionary> createState() => \_DictionaryState();

}

class \_DictionaryState extends State<Dictionary> {

  Icon customIcon = const Icon(Icons.search);

  Widget customSearchBar = const Text('Dictionary');

  dynamic \_jsonData;

  static List<Word> wordList = [];

  List<Word> searchedWordList = List.from(wordList);

  ScrollController controller = new ScrollController();

  TextEditingController \_seachController = new TextEditingController();

  @override

  void initState() {

    super.initState();

    widget.storage.loadDefautWords(context).then((dynamic json) {

      setState(() {

        \_jsonData = json;

        \_jsonData.keys.forEach((key) {

          wordList.add(Word(key, \_jsonData[key]));

        });

        searchedWordList = wordList;

      });

    });

  }

  @override

  void dispose() {

    super.dispose();

  }

  onItemChanged(String value) {

    setState(() {

      //tim chinh xac

      searchedWordList = wordList

          .where((w) => w.key.toLowerCase() == value.toLowerCase())

          .toList();

      //tim gan dung

      if (searchedWordList.length == 0) {

        searchedWordList = wordList

            .where((w) =>

                w.key.toLowerCase().contains(value.toLowerCase()) ||

                w.value.toLowerCase().contains(value.toLowerCase()))

            .toList();

      }

    });

  }

  Widget buildSeach() {

    return TextField(

      controller: \_seachController,

      decoration: InputDecoration(

          fillColor: Colors.white,

          filled: true,

          hintText: 'Search...',

          contentPadding: EdgeInsets.all(15),

          border: InputBorder.none),

      onChanged: onItemChanged,

    );

  }

  Widget buildWordList() {

    return Expanded(

        child: searchedWordList.length == 0

            ? Text("Không tìm thấy")

            : ListView.builder(

                controller: controller,

                itemCount: searchedWordList.length,

                itemBuilder: (BuildContext context, int i) {

                  return ListTile(

                    onTap: () {

                      //qua trang chi tiet tu

                      Navigator.push(

                        context,

                        MaterialPageRoute(

                            builder: (context) => WordDetail(

                                  word: searchedWordList[i],

                                )),

                      );

                    },

                    title: Text(searchedWordList[i].key),

                    subtitle: Text(searchedWordList[i].value.length < 20

                        ? searchedWordList[i].value

                        : searchedWordList[i].value.substring(0, 20) + "..."),

                  );

                }));

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: customSearchBar,

        automaticallyImplyLeading: false,

        centerTitle: true,

        actions: [

          Container(

            width: 100,

            child: IconButton(

              onPressed: () {

                setState(() {

                  if (customIcon.icon == Icons.search) {

                    customIcon = const Icon(Icons.cancel);

                    customSearchBar = buildSeach();

                  } else if (customIcon.icon == Icons.cancel) {

                    customIcon = const Icon(Icons.search);

                    customSearchBar = Text("Dictionary");

                  }

                });

              },

              icon: customIcon,

            ),

          )

        ],

      ),

      body: Container(

          child: Column(

        children: [

          buildWordList(),

        ],

      )),

      floatingActionButton: FloatingActionButton.extended(

        onPressed: () {

          Navigator.pushNamed(context, '/add\_word');

        },

        label: const Text('Add Word'),

        icon: const Icon(Icons.add),

      ),

    );

  }

}