product dart

// import 'dart:ffi';

import 'dart:ui';

import 'package:flutter/material.dart';

import 'package:sliding\_up\_panel/sliding\_up\_panel.dart';

// import 'package:flutter/services.dart';

import 'package:cached\_network\_image/cached\_network\_image.dart';

import 'package:flutter\_svg/flutter\_svg.dart';

import 'package:carousel\_slider/carousel\_slider.dart';

import 'imgPreview.dart';

// import 'test.dart';

import 'package:fab\_circular\_menu/fab\_circular\_menu.dart';

import 'package:provider/provider.dart';

import 'package:myFirst/models/search\_model.dart';

import 'package:like\_button/like\_button.dart';

import 'package:flutter\_spinkit/flutter\_spinkit.dart';

import 'package:myFirst/easy\_request.dart';

class Product extends StatelessWidget {

final Map<String, dynamic> itemData;

Product({Key key, @required this.itemData}) : super(key: key);

List bookImg;

final double \_initFabHeight = 120.0;

static const Map labelTitle = {

"isbn": "ISBN",

"page": "Pages",

"pub": "Publisher",

"author": "Author",

"date": "Publication Date",

"price": "Price"

};

// static const Map labelTitle = {

// "title": "",

// "isbn": "",

// "page": "",

// "pub": "",

// "author": "",

// "date": "",

// "price": ""

// };

static const Map tabTitle = {"intr": "簡介", "author": "作者介紹", "content": "目錄"};

// Dialog

List<Widget> indexFrame = [SizedBox(), SizedBox(), SizedBox()];

List<RaisedButton> cityBtnFrame = [];

List<RaisedButton> univBtnFrame = [];

Map<String, Map> universityData = {

"taipei": {

"title": "臺北",

"info": {

"mkc": {"title": "馬偕醫護管理專科學校"},

"nccu": {"title": "國立政治大學"},

"ntu": {"title": "國立臺灣大學"},

"ntnu": {"title": "國立臺灣師範大學"},

"ym": {"title": "國立陽明大學"},

"ntust": {"title": "國立臺灣科技大學"},

"ntut": {"title": "國立臺北科技大學"},

"tnua": {"title": "國立臺北藝術大學"},

"ntue": {"title": "國立臺北教育大學"},

"ntunhs": {"title": "國立臺北護理健康大學"},

"ntub": {"title": "國立臺北商業大學"},

"scu": {"title": "東吳大學"},

"pccu": {"title": "中國文化大學"},

"shu": {"title": "世新大學"},

"mcu": {"title": "銘傳大學"},

"usc": {"title": "實踐大學"},

"ttu": {"title": "大同大學"},

"tmu": {"title": "臺北醫學大學"},

"cute": {"title": "中國科技大學"},

"takming": {"title": "德明財經科技大學"},

"cust": {"title": "中華科技大學"},

"tpcu": {"title": "臺北城市科技大學"},

"ukn": {"title": "康寧大學"},

"uTaipei": {"title": "臺北市立大學"},

"tcpa": {"title": "國立臺灣戲曲學院"},

},

},

"hsinchu": {

"title": "新竹",

"info": {

"ypu": {"title": "元培醫事科技大學"},

"nthu": {"title": "國立清華大學"},

"nctu": {"title": "國立交通大學"},

"chu": {"title": "中華大學"},

"hcu": {"title": "玄奘大學"},

"tust": {"title": "敏實科技大學"},

"must": {"title": "明新科技大學"}

},

},

"tainan ": {

"title": "臺南",

"info": {

"mhchcm": {"title": "敏惠醫護管理專科學校"},

"ncku": {"title": "國立成功大學"},

"tnnua": {"title": "國立臺南藝術大學"},

"nutn": {"title": "國立臺南大學"},

"stust": {"title": "南臺科技大學"},

"ksu": {"title": "崑山科技大學"},

"cnu": {"title": "嘉南藥理大學"},

"cjcu": {"title": "長榮大學"},

"tut": {"title": "台南應用科技大學"},

"feu": {"title": "遠東科技大學"},

"hwai": {"title": "中華醫事科技大學"},

"tsu": {"title": "台灣首府大學"},

"ctbc": {"title": "中信金融管理學院"},

"ntin": {"title": "國立臺南護理專科學校"},

},

},

"taichung ": {

"title": "臺中",

"info": {

"hust": {"title": "修平科技大學"},

"nchu": {"title": "國立中興大學"},

"ntcu": {"title": "國立臺中教育大學"},

"ncut": {"title": "國立勤益科技大學"},

"ntupes": {"title": "國立臺灣體育運動大學"},

"nutc": {"title": "國立臺中科技大學"},

"thu": {"title": "東海大學"},

"fcu": {"title": "逢甲大學"},

"pu": {"title": "靜宜大學"},

"cyut": {"title": "朝陽科技大學"},

"csmu": {"title": "中山醫學大學"},

"hk": {"title": "弘光科技大學"},

"cmu": {"title": "中國醫藥大學"},

"ltu": {"title": "嶺東科技大學"},

"ctust": {"title": "中臺科技大學"},

"asia": {"title": "亞洲大學"},

"ocu": {"title": "僑光科技大學"},

},

},

"taoyuan": {

"title": "桃園",

"info": {

"hsc": {"title": "新生醫護管理專科學校"},

"ncu": {"title": "國立中央大學"},

"ntsu": {"title": "國立體育大學"},

"cycu": {"title": "中原大學"},

"cgu": {"title": "長庚大學"},

"yzu": {"title": "元智大學"},

"lhu": {"title": "龍華科技大學"},

"uch": {"title": "健行科技大學"},

"vnu": {"title": "萬能科技大學"},

"knu": {"title": "開南大學"},

"cgust": {"title": "長庚科技大學"},

"nanya": {"title": "南亞技術學院"},

},

},

"kaohsiung ": {

"title": "高雄",

"info": {

"yuhing": {"title": "育英醫護管理專科學校"},

"nsysu": {"title": "國立中山大學"},

"nknu": {"title": "國立高雄師範大學"},

"nuk": {"title": "國立高雄大學"},

"nkuht": {"title": "國立高雄餐旅大學"},

"nkust": {"title": "國立高雄科技大學"},

"isu": {"title": "義守大學"},

"kmu": {"title": "高雄醫學大學"},

"stu": {"title": "樹德科技大學"},

"fy": {"title": "輔英科技大學"},

"csu": {"title": "正修科技大學"},

"kyu": {"title": "高苑科技大學"},

"wzu": {"title": "文藻外語大學"},

"tf": {"title": "東方設計大學"},

"fotech": {"title": "和春技術學院"},

"szmc": {"title": "樹人醫護管理專科學校"},

},

},

"keelung": {

"title": "基隆",

"info": {

"cku": {"title": "經國管理暨健康學院"},

"ntou": {"title": "國立臺灣海洋大學"},

"cufa": {"title": "崇右影藝科技大學"},

},

},

"changhua ": {

"title": "彰化",

"info": {

"ccut": {"title": "中州科技大學"},

"ncue": {"title": "國立彰化師範大學"},

"dyu": {"title": "大葉大學"},

"ctu": {"title": "建國科技大學"},

"mdu": {"title": "明道大學"},

},

},

"newTaipei": {

"title": "新北",

"info": {

"ctcn": {"title": "耕莘健康管理專科學校"},

"ntpu": {"title": "國立臺北大學"},

"ntua": {"title": "國立臺灣藝術大學"},

"fju": {"title": "輔仁大學"},

"tku": {"title": "淡江大學"},

"hfu": {"title": "華梵大學"},

"au": {"title": "真理大學"},

"mcut": {"title": "明志科技大學"},

"sju": {"title": "聖約翰科技大學"},

"just": {"title": "景文科技大學"},

"tnu": {"title": "東南科技大學"},

"hwu": {"title": "醒吾科技大學"},

"hwh": {"title": "華夏科技大學"},

"chihlee": {"title": "致理科技大學"},

"hdut": {"title": "宏國德霖科技大學"},

"tumt": {"title": "台北海洋科技大學"},

"oit": {"title": "亞東技術學院"},

"lit": {"title": "黎明技術學院"},

"mmc": {"title": "馬偕醫學院"},

"dila": {"title": "法鼓文理學院"},

},

},

"chiayi": {

"title": "嘉義",

"info": {

"ttc": {"title": "大同技術學院"},

"ncyu": {

"title": "國立嘉義大學",

"cjc": {"title": "崇仁醫護管理專科學校"},

"ccu": {"title": "國立中正大學"},

"nhu": {"title": "南華大學"},

"wfu": {"title": "吳鳳科技大學"},

"toko": {"title": "稻江科技暨管理學院"},

}

},

},

"hualien": {

"title": "花蓮",

"info": {

"tht": {"title": "臺灣觀光學院"},

"ndhu": {"title": "國立東華大學"},

"tcu": {"title": "慈濟大學"},

"tcust": {"title": "慈濟科技大學"},

"dahan": {"title": "大漢技術學院"},

},

},

"nantou ": {

"title": "南投",

"info": {

"nkut": {"title": "南開科技大學"},

"ncnu": {"title": "國立暨南國際大學"}

},

},

"yunlin": {

"title": "雲林",

"info": {

"twu": {"title": "環球科技大學"},

"yuntech": {"title": "國立雲林科技大學"},

"nfu": {"title": "國立虎尾科技大學"},

},

},

"pingtung": {

"title": "屏東",

"info": {

"tzuhui": {"title": "慈惠醫護管理專科學校"},

"npust": {"title": "國立屏東科技大學"},

"nptu": {"title": "國立屏東大學"},

"tajen": {"title": "大仁科技大學"},

"meiho": {"title": "美和科技大學"},

},

},

"taitung": {

"title": "臺東",

"info": {

"ntc": {"title": "國立臺東專科學校"},

"nttu": {"title": "國立臺東大學"}

},

},

"yilan": {

"title": "宜蘭",

"info": {

"smc": {"title": "聖母醫護管理專科學校"},

"niu": {"title": "國立宜蘭大學"},

"fgu": {"title": "佛光大學"},

"fit": {"title": "蘭陽技術學院"},

},

},

"miaoli": {

"title": "苗栗",

"info": {

"jente": {"title": "仁德醫護管理專科學校"},

"nuu": {"title": "國立聯合大學"},

"ydu": {"title": "育達科技大學"},

},

},

"penghu ": {

"title": "澎湖",

"info": {

"npu": {"title": "國立澎湖科技大學"}

}

},

"kinmen": {

"title": "金門",

"info": {

"nqu": {"title": "國立金門大學"}

}

},

};

double \_fabHeight;

double \_panelHeightOpen;

double \_panelHeightClosed = 95.0;

CarouselController buttonCarouselController = CarouselController(); // 圖片滑動

final GlobalKey<FabCircularMenuState> fabKey = GlobalKey(); // 浮動按鈕2

final CarouselController \_imgController = CarouselController();

var imgCurrentPage = 0;

// @override

// void initState() {

// super.initState();

// \_fabHeight = \_initFabHeight;

// }

// --- item information ---

Widget labelInfo({String img, String title, Color color}) => ListTile(

// isThreeLine: true,

leading: Container(

width: 24,

child: Align(

alignment: Alignment.centerLeft,

child: SvgPicture.asset(

"assets/$img.svg",

width: 22,

// color: color,

),

),

),

// leading: Text(img, style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),),

title: Text(title, style: TextStyle(fontSize: 14, color: Colors.black)),

subtitle: Text("# ${labelTitle[img]}"),

enabled: false,

// shape: ,

// onLongPress: ()=>show,

);

Widget test({String title, String text}) {

return ListView(

// controller: sc,

children: [

Container(

padding: const EdgeInsets.symmetric(horizontal: 15, vertical: 25),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: <Widget>[

Text(title,

style: TextStyle(fontWeight: FontWeight.w600, fontSize: 18)),

SizedBox(

height: 12.0,

),

Text(

text,

softWrap: true,

),

SizedBox(

height: 20.0,

),

],

),

),

],

);

}

@override

Widget build(BuildContext context) {

print(List.generate(itemData['intr'].length, (index) => index \* 5));

\_panelHeightOpen = MediaQuery.of(context).size.height \* .80;

bookImg = itemData["imgURL"]

.map(

(item) => CachedNetworkImage(

imageUrl:

"https://${item.substring(0, 3)}.book.com.tw/image/getImage?i=https://www.books.com.tw/img/${item.substring(3)}",

// "https://im1.book.com.tw/image/getImage?i=https://www.books.com.tw/img/001/086/84/0010868410.jpg&v=5f3e50aa&w=348&h=348",

fit: BoxFit.fitHeight,

placeholder: (BuildContext context, String url) => const Center(

// child: CircularProgressIndicator(),

child: Icon(Icons.photo\_size\_select\_actual),

),

errorWidget: (BuildContext context, String url, dynamic error) =>

const Center(child: Icon(Icons.error)),

fadeOutDuration: const Duration(seconds: 1),

fadeInDuration: const Duration(seconds: 2),

// placeholder: ,

),

)

.toList();

return ChangeNotifierProvider(

create: (\_) => SearchModel(),

// create: (\_) => searchNotifier,

child: Scaffold(

backgroundColor: Colors.white,

appBar: AppBar(

elevation: 1,

backgroundColor: Colors.white,

leading: BackButton(color: Colors.black),

actions: [

PopupMenuButton<String>(

icon: Icon(Icons.more\_vert, color: Colors.black),

itemBuilder: (BuildContext context) => <PopupMenuItem<String>>[

PopupMenuItem<String>(

value: 'value02',

child: Row(children: <Widget>[

Padding(

padding: EdgeInsets.only(right: 10),

child: Icon(

Icons.share,

color: Colors.black,

)),

Text('Share')

])),

// PopupMenuItem<String>(

// value: 'value01', child: Text('other')),

// PopupMenuItem<String>(

// value: 'value02', child: Text('Item Two')),

],

onSelected: (String value) {

showSchoolDialog();

})

],

),

body: Stack(

alignment: Alignment.topCenter,

children: <Widget>[

SlidingUpPanel(

maxHeight: \_panelHeightOpen,

minHeight: \_panelHeightClosed,

// parallaxEnabled: true,

parallaxOffset: .5,

panel: \_panel(context),

collapsed: Center(

child: Text(

"商品資訊",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 18.0,

letterSpacing: 3.0,

color: Colors.grey[600]),

),

),

body: \_body(context),

borderRadius: BorderRadius.only(

topLeft: Radius.circular(18.0),

topRight: Radius.circular(18.0)),

// onPanelSlide: (double pos) => setState(() {

// \_fabHeight = pos \* (\_panelHeightOpen - \_panelHeightClosed) +

// \_initFabHeight;

// }),

),

],

),

floatingActionButton: Builder(

builder: (context) => FabCircularMenu(

key: fabKey,

// Cannot be `Alignment.center`

alignment: Alignment.bottomRight,

ringColor: Colors.black.withAlpha(150),

ringDiameter: 280.0,

ringWidth: 80.0,

fabSize: 54.0, // 浮動按鈕大小

fabElevation: 10.0,

fabMargin: const EdgeInsets.all(20.0),

// Also can use specific color based on wether

// the menu is open or not:

// fabOpenColor: Colors.white

// fabCloseColor: Colors.white

// These properties take precedence over fabColor

fabColor: Colors.white,

fabOpenIcon: Icon(Icons.menu, color: Colors.black),

fabCloseIcon: Icon(Icons.close, color: Colors.black),

animationDuration: const Duration(milliseconds: 800),

animationCurve: Curves.easeInOutCirc,

// onDisplayChange: (isOpen) {

// \_showSnackBar(

// context, "The menu is ${isOpen ? "open" : "closed"}");

// },

children: <Widget>[

CircleAvatar(

backgroundColor: Colors.white,

child: LikeButton(

onTap: onLikeButtonTapped, //jsonData["index"]

size: 30,

bubblesColor: BubblesColor(

dotPrimaryColor: Color(0xFFB933),

dotSecondaryColor: Color(0xFFEA33),

),

likeBuilder: (bool isLiked) {

return Icon(

Icons.star, // 圖案

color: isLiked ? Colors.yellow[600] : Colors.grey,

);

},

),

),

RawMaterialButton(

shape: CircleBorder(),

padding: const EdgeInsets.all(10.0),

fillColor: Colors.white,

child: Icon(Icons.school, color: Colors.black),

onPressed: () => showSchoolDialog(

context: context,

title: itemData['title'],

isbn: itemData['ISBN']),

),

Consumer<SearchModel>(

builder: (\_, localNotify, \_\_) => RawMaterialButton(

shape: CircleBorder(),

padding: const EdgeInsets.all(10.0),

fillColor: Colors.white,

child: Icon(Icons.add, color: Colors.grey),

onPressed: () => localNotify.changeDialogPage(0),

),

),

// RawMaterialButton(

// shape: CircleBorder(),

// padding: const EdgeInsets.all(10.0),

// fillColor: Colors.white,

// child: Icon(Icons.link, color: Colors.black),

// onPressed: () {},

// ),

],

),

),

// floatingActionButtonLocation: FloatingActionButtonLocation.,

),

);

}

Widget \_panel(BuildContext context) {

return MediaQuery.removePadding(

context: context,

removeTop: true,

child: Column(

children: <Widget>[

SizedBox(

height: 12.0,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Container(

width: 30,

height: 5,

decoration: BoxDecoration(

color: Colors.grey[300],

borderRadius: BorderRadius.all(Radius.circular(12.0))),

),

],

),

SizedBox(

height: 60.0,

),

Expanded(

child: DefaultTabController(

length: itemData['intr'].length,

child: Scaffold(

// backgroundColor: Colors.transparent,

appBar: AppBar(

backgroundColor: Colors.white,

title: Center(

child: TabBar(

isScrollable: true,

// indicatorWeight: 2,

// indicatorPadding:

// EdgeInsets.symmetric(horizontal: 20, vertical: 5),

labelPadding: EdgeInsets.symmetric(horizontal: 30),

indicatorColor: Colors.grey[700],

labelColor: Colors.black,

tabs: itemData['intr']

.map<Widget>((item) => Tab(

text: tabTitle[item['title']],

))

.toList(),

),

),

),

body: TabBarView(

children: List.generate(

itemData['intr'].length,

(index) => test(

// sc: sc,

title: tabTitle[itemData['intr'][index]['title']],

text: itemData['intr'][index]['content']))),

),

),

),

],

),

);

}

Widget \_body(BuildContext context) {

return Container(

padding: const EdgeInsets.only(bottom: 130),

child: ListView(

children: [

SizedBox(height: 30),

Stack(

children: [

Consumer<SearchModel>(

builder: (context, localNotify, \_) => CarouselSlider(

carouselController: \_imgController,

options: CarouselOptions(

aspectRatio: 16 / 9,

viewportFraction: 0.65,

initialPage: 0,

enableInfiniteScroll: true,

reverse: false,

onPageChanged: (index, \_) =>

localNotify.switchImagePage(index),

// autoPlay: true,

// autoPlayInterval: Duration(seconds: 1),

// autoPlayAnimationDuration: Duration(milliseconds: 500),

// autoPlayCurve: Curves.easeIn,

enlargeCenterPage: true,

// onPageChanged: callbackFunction,

scrollDirection: Axis.horizontal,

),

items: List.generate(

bookImg.length,

(index) => GestureDetector(

onTap: () => Navigator.push(

context,

MaterialPageRoute(

builder: (context) => BookImgPreview(

index: index,

imgList: bookImg,

notify: localNotify),

)),

// onTap: () => print(item.fit),

child: Center(child: bookImg[index]

// Image.network(item, fit: BoxFit.fitHeight)

),

))),

),

Positioned(

bottom: 0,

right: 25,

child: Container(

width: 60,

padding: EdgeInsets.symmetric(vertical: 7),

decoration: BoxDecoration(

color: Colors.grey[600],

borderRadius: BorderRadius.all(Radius.circular(12.0))),

child: Center(

child: Selector<SearchModel, int>(

builder: (context, title, child) => Text(

'$title / ${bookImg.length}',

style: TextStyle(color: Colors.white),

),

selector: (context, viewModel) =>

viewModel.currentPage)),

),

)

],

),

Divider(),

Padding(

padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),

child: Text(

itemData['info']['title'],

style: TextStyle(fontSize: 18),

),

),

Divider(),

Padding(

padding: const EdgeInsets.fromLTRB(15, 0, 0, 80),

child: Column(

children: itemData['info'].entries.map<Widget>((entry) {

return entry.key != 'title'

? labelInfo(

img: entry.key,

title: entry.value,

color: Color(0xFF0000))

: SizedBox(

height: 0,

);

// if (entry.key != 'title'){

// return labelInfo(

// img: entry.key,

// title: entry.value,

// color: Color(0xFF0000));

// }

}).toList(),

)),

],

),

);

}

Future<bool> onLikeButtonTapped(bool isLiked) async {

/// send your request here

// final bool success= await sendRequest();

/// if failed, you can do nothing

// return success? !isLiked:isLiked;

return !isLiked;

}

// --- Show Dailog ---

Future searchUnivBookStatus({String city, String univ}) async {

String url = '127.0.0.1:8000/book/test/';

var data = await fetchAlbum(url);

return data;

}

void showSchoolDialog({BuildContext context, String title, int isbn}) {

showDialog(

context: context,

builder: (BuildContext context) {

return SimpleDialog(

titlePadding: EdgeInsets.only(top: 24),

title: Center(child: Text("學校搜尋")),

contentPadding: EdgeInsets.all(32),

children: [

SizedBox(

height: 10,

),

Consumer<SearchModel>(builder: (\_, localNotify, \_\_) {

// localNotify.initDialog();

Widget dailogBar() {

return Align(

alignment: Alignment.centerLeft,

child: IconButton(

icon: Icon(Icons.arrow\_back),

onPressed: () {

switch (localNotify.dialogPageCount) {

case 0:

Navigator.pop(context);

break;

case 1:

localNotify.changeDialogPage(0);

break;

case 2:

localNotify.changeDialogPage(1);

break;

default:

}

}),

);

}

Widget btnsFrame({List<Widget> btnList}) {

return Center(

child: Column(

children: [

dailogBar(),

Wrap(

direction: Axis.horizontal,

spacing: 15.0,

runSpacing: 4.0,

children: btnList),

// RaisedButton(

// child: Text('back'),

// onPressed: () => localNotify.changeDialogPage(0)),

],

),

);

}

Widget searchFrame({String title, String city, String univ}) {

// var data = searchUnivBookStatus();

return Column(

children: [

Row(

children: [

dailogBar(),

Center(

child: Card(

child: Padding(

padding: const EdgeInsets.all(10.0),

child: Text(title),

),

),

)

],

),

Text("查找 $title"),

Text('Loading...'),

Text('$city...'),

Text('$univ...')

],

);

}

// make university buttons function

Future buildUnivBtns({String cityKey}) async {

print('buildUnivBtns');

universityData[cityKey]['info'].forEach((key, value) {

univBtnFrame.add(RaisedButton(

color: Colors.white,

child: Text(value['title']),

onPressed: () {

print('click2');

localNotify.seleTags['univ'] = key;

indexFrame[2] = searchFrame(

title: value['title'], city: cityKey, univ: key);

localNotify.changeDialogPage(2);

},

));

});

}

// init cityBtnFrame make buttons if list data is empty

if (cityBtnFrame.length == 0) {

print('cityBtnFrame.length: ${cityBtnFrame.length}');

universityData

.forEach((key, value) => cityBtnFrame.add(RaisedButton(

color: Colors.white,

child: Text(value['title']),

onPressed: () async {

print('click1');

localNotify.seleTags['city'] = key;

await buildUnivBtns(cityKey: key);

indexFrame[1] = btnsFrame(btnList: univBtnFrame);

print(indexFrame[1]);

localNotify.changeDialogPage(1);

},

)));

indexFrame[0] = btnsFrame(btnList: cityBtnFrame);

}

print(cityBtnFrame.length);

return IndexedStack(

index: localNotify.dialogPageCount, children: indexFrame);

}),

// Selector<SearchModel, Map>(

// selector: (\_, viewModel) => viewModel.seleTags,

// builder: (buildContext2, seleTags, widget2) {

// print('1');

// return Selector<SearchModel, (Map, int)>(

// selector: (context, viewModel) =>

// viewModel.dialogPageCount,

// builder: (\_, dialogPageCount, \_\_) {

// List<Widget> indexFrame = [null, Text('x')];

// List<RaisedButton> cityBtnFrame = [];

// List<RaisedButton> univBtnFrame = [];

// Widget btnsFrame({List<Widget> btnList}) => Wrap(

// direction: Axis.horizontal,

// spacing: 15.0,

// runSpacing: 4.0,

// children: btnList);

// // make university buttons function

// Future makeUnivBtns({String localKey}) {

// universityData[localKey]['info'].forEach(

// (key, value) =>

// univBtnFrame.add(RaisedButton(

// color: Colors.white,

// child: Text(value['title']),

// onPressed: () {

// seleTags['univ'] = key;

// dialogPageCount = 0;

// print(dialogPageCount);

// },

// )));

// }

// // init cityBtnFrame make buttons

// universityData.forEach(

// (key, value) => cityBtnFrame.add(RaisedButton(

// color: Colors.white,

// child: Text(value['title']),

// onPressed: () async {

// seleTags['city'] = key;

// indexFrame[1] = await makeUnivBtns(localKey: key);

// dialogPageCount = 1;

// print(seleTags);

// print(dialogPageCount);

// },

// )));

// indexFrame[0] = btnsFrame(btnList: cityBtnFrame);

// return IndexedStack(

// index: dialogPageCount,

// children: indexFrame);

// },

// );

// }))

],

);

});

}

}