import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:flutter/material.dart';

import 'package:hdtc\_project/services/firestore\_services.dart';

import 'package:hdtc\_project/utils.dart';

class UniversitySpecializationScreen extends StatefulWidget {

// final List<Map> spec;

final String uniName;

final String branch;

final String field;

const UniversitySpecializationScreen({

Key? key,

required this.field,

required this.uniName,

required this.branch,

}) : super(key: key);

@override

\_UniversitySpecializationScreenState createState() =>

\_UniversitySpecializationScreenState();

}

class \_UniversitySpecializationScreenState

extends State<UniversitySpecializationScreen> {

bool isDesktop(BuildContext context) {

return MediaQuery.of(context).size.width >= 600;

}

@override

Widget build(BuildContext context) {

return StreamBuilder<DocumentSnapshot>(

stream: FireStoreServices.getUniData(

branch: widget.branch, uniName: widget.uniName),

builder: (context, snapshot) {

switch (snapshot.connectionState) {

case ConnectionState.waiting:

return Center(

child: CircularProgressIndicator(),

);

case ConnectionState.active:

if (snapshot.hasError) {

return Center(

child: Text('Something Went Wrong please try again later'),

);

} else {

final data = snapshot.data;

final name = data!['name'];

final specs = List.of(data['specializations'])

.where((element) => element.values.contains(widget.field))

.toList();

return GridView.builder(

padding: EdgeInsets.all(16),

gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

childAspectRatio: 1,

crossAxisCount: isDesktop(context) ? 4 : 2,

crossAxisSpacing: 25,

mainAxisSpacing: 25),

itemBuilder: (context, index) {

return Container(

color: Colors.grey,

padding: EdgeInsets.all(16),

child: DefaultTextStyle(

style: TextStyle(

fontSize: 20, fontWeight: FontWeight.normal),

child: GridTile(

header: FittedBox(

child: Text.rich(

TextSpan(children: [

TextSpan(

text: 'Field: ',

style: TextStyle(fontWeight: FontWeight.bold),

),

TextSpan(

style: TextStyle(fontSize: 18),

text: '${specs[index]['field']}',

),

]),

textAlign: TextAlign.center,

),

),

footer: Center(

child: Material(

color: Colors.grey,

child: IconButton(

onPressed: () {}, icon: Icon(Icons.edit)),

),

),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

FittedBox(

child: Text.rich(

TextSpan(children: [

TextSpan(

text: 'Spec: ',

style: TextStyle(

fontWeight: FontWeight.bold),

),

TextSpan(

text: Utils.capitalizeFirstOfEachWord(

'${specs[index]['spec']}\n'),

style: TextStyle(fontSize: 18),

),

TextSpan(

text: 'Lang: ',

style: TextStyle(

fontWeight: FontWeight.bold),

),

TextSpan(

text: Utils.capitalizeFirstOfEachWord(

'${specs[index]['lang']}\n'),

style: TextStyle(fontSize: 18),

),

TextSpan(

text: 'Fees: ',

style: TextStyle(

fontWeight: FontWeight.bold),

),

TextSpan(

text: Utils.capitalizeFirstOfEachWord(

'${specs[index]['fees']}\n'),

style: TextStyle(fontSize: 18),

),

TextSpan(

text: 'Location: ',

style: TextStyle(

fontWeight: FontWeight.bold),

),

TextSpan(

text: Utils.capitalizeFirstOfEachWord(

'${specs[index]['location']}\n'),

style: TextStyle(fontSize: 18),

),

TextSpan(

text: 'Note: ',

style: TextStyle(

fontSize: 20,

fontWeight: FontWeight.bold),

),

TextSpan(

style: TextStyle(fontSize: 18),

text: Utils.capitalizeFirstOfEachWord(

'${specs[index]['note']}\n'),

),

]),

textAlign: TextAlign.center,

),

),

],

),

),

),

);

},

itemCount: specs.length,

);

}

default:

return Center(

child: Text('Default Case'),

);

}

final data = snapshot.data;

return Scaffold();

});

}

}