import 'dart:math';  
  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter\_screenutil/flutter\_screenutil.dart';  
import 'package:insuranceadminside/UserDetails.dart';  
  
class Search extends StatefulWidget {  
 const Search({super.key});  
  
 @override  
 State<Search> createState() => \_SearchState();  
}  
  
class \_SearchState extends State<Search> {  
 TextEditingController searchcontroller = TextEditingController();  
 final firestore = FirebaseFirestore.*instance*.collection("Users").snapshots();  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 backgroundColor: Colors.*grey*.withOpacity(0.4),  
 ),  
 body: Container(  
 width: 1440.w, // Full container width  
 height: 800.h, // Full container height  
 decoration: BoxDecoration(  
 image: DecorationImage(  
 image: AssetImage("assets/img.jpg"), // Background image  
 fit: BoxFit.cover, // Full-screen image  
 ),  
 ),  
 child: Container(  
 color: Colors.*grey*.withOpacity(0.4),  
 child: SingleChildScrollView(  
 child: Column(  
 children: [  
 Padding(  
 padding: EdgeInsets.only(left: 60.w, top: 20.h, right: 60.w),  
 child: TextField(  
 controller: searchcontroller,  
 decoration: InputDecoration(  
 hintText: "Search by Name, Registration ID,Vehicle Number , or Phone...",  
 prefixIcon: Icon(Icons.*search*),  
 border: OutlineInputBorder()),  
 onChanged: (text) {  
 setState(() {}); // Trigger a rebuild when text changes  
 },  
 ),  
 ),  
 StreamBuilder<QuerySnapshot>(  
 stream: firestore,  
 builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {  
 if (!snapshot.hasData) {  
 return Center(  
 child: CircularProgressIndicator(),  
 );  
 }  
 if (snapshot.hasError) {  
 return Center(  
 child: Text(  
 "ERROR",  
 style: TextStyle(color: Colors.*red*),  
 ),  
 );  
 }  
 if (snapshot.hasData) {  
 // Get the search query from the controller  
 String searchText = searchcontroller.text.toLowerCase();  
  
 // Filter the documents based on the search query (name, regId, phone)  
 var filteredDocs = snapshot.data!.docs.where((doc) {  
 // Case insensitive comparison for all 3 fields  
 String name = doc["name"].toString().toLowerCase();  
 String regId = doc["regId"].toString().toLowerCase();  
 String phone = doc["phone"].toString().toLowerCase();  
 String vehiclenumber = doc["vehicleNumber"].toString().toLowerCase();  
  
 return name.contains(searchText) || regId.contains(searchText) || phone.contains(searchText) || vehiclenumber.contains(searchText);  
 }).toList();  
  
 // If no matching documents are found  
 if (filteredDocs.isEmpty) {  
 return Center(  
 child: Text("No results found"),  
 );  
 }  
  
 return SizedBox(  
 width: 1440.w,  
 height: max(260.h, 100 \* snapshot.data!.docs.length.h),  
 child: GridView.builder(  
 itemCount: filteredDocs.length,  
 physics: NeverScrollableScrollPhysics(),  
 gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
 crossAxisCount: 5,  
 crossAxisSpacing: 10.0,  
 mainAxisSpacing: 10.0,  
 ),  
 itemBuilder: (context, index) {  
 var doc = filteredDocs[index];  
 return Padding(  
 padding: EdgeInsets.all(10.0.sp),  
 child: InkWell(onTap: (){  
  
  
 Navigator.*of*(context).push(MaterialPageRoute(builder: (\_)=> Userdetails(Id: doc['docId'])));  
  
 },  
 child: Container(  
 width: 100.w,  
 height: 100.h,  
 decoration: ShapeDecoration(  
 color: Colors.*lightBlue*[100],  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(10.r))),  
 child: Padding(  
 padding: EdgeInsets.only(left: 10.w, right: 10.w, top: 10.h),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Row(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Image.asset(  
 "assets/profile.jpg",  
 fit: BoxFit.cover,  
 width: 80.w,  
 height: 110.h,  
 ),  
 SizedBox(  
 width: 10.w,  
 ),  
 Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(  
 doc["name"],  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 20.sp,  
 fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 height: 10.h,  
 ),  
 Text(  
 doc["regId"],  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 20.sp,  
 fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 height: 10.h,  
 ),  
 Text(  
 doc["phone"],  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 20.sp,  
 fontWeight: FontWeight.*bold*),  
 ),  
 ],  
 )  
 ],  
 ),  
 Text(  
 "DEVICE ID :${doc["deviceId"]}",  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 18.sp,  
 fontWeight: FontWeight.*bold*,  
 ),  
 maxLines: 1,  
 overflow: TextOverflow.ellipsis,  
 ),  
 SizedBox(  
 height: 10.h,  
 ),  
 Text(  
 "TYPE : ${doc["InsuranceType"]}",  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 18.sp,  
 fontWeight: FontWeight.*bold*,  
 ),  
 maxLines: 1,  
 overflow: TextOverflow.ellipsis,  
 ),  
 SizedBox(  
 height: 10.h,  
 ),  
 Text(  
 doc["address"],  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 18.sp,  
 fontWeight: FontWeight.*bold*,  
 ),  
 maxLines: 2,  
 overflow: TextOverflow.ellipsis,  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 );  
 },  
 ),  
 );  
 } else {  
 return SizedBox();  
 }  
 },  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
}