import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter\_screenutil/flutter\_screenutil.dart';  
import 'package:image\_picker/image\_picker.dart';  
import 'dart:io'; // For File type  
import 'dart:convert'; // For base64 encoding  
import 'package:flutter/foundation.dart';  
  
class Claimpage extends StatefulWidget {  
 final String id;  
 const Claimpage({super.key, required this.id});  
  
 @override  
 State<Claimpage> createState() => \_ClaimpageState();  
}  
  
class \_ClaimpageState extends State<Claimpage> {  
 String? \_selectedValue;  
 File? \_image;  
 String? \_imageBase64;  
 TextEditingController hpnamecontroller = TextEditingController();  
 TextEditingController hpnumbercontroller = TextEditingController();  
 TextEditingController hpipnumbercontroller = TextEditingController();  
 TextEditingController firnumbercontroller = TextEditingController();  
 TextEditingController billnumbercontroller = TextEditingController();  
 TextEditingController amountcontroller = TextEditingController();  
 TextEditingController patientnamecontroller = TextEditingController();  
  
 String? insuranceType;  
 final FirebaseFirestore \_firestore = FirebaseFirestore.*instance*;  
 final ImagePicker \_picker = ImagePicker();  
 final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();  
  
  
  
  
 Future<void> \_pickImage() async {  
 final XFile? pickedFile = await \_picker.pickImage(source: ImageSource.gallery);  
 if (pickedFile != null) {  
 if (kIsWeb) {  
 pickedFile.readAsBytes().then((bytes) {  
 setState(() {  
 \_imageBase64 = base64Encode(bytes);  
 });  
 });  
 } else {  
 setState(() {  
 \_image = File(pickedFile.path);  
 });  
 }  
 }  
 }  
  
 // Fetch user's insurance type  
 Future<void> \_fetchInsuranceType() async {  
 try {  
 DocumentSnapshot userDoc = await \_firestore.collection('Users').doc(widget.id).get();  
 if (userDoc.exists) {  
 setState(() {  
 insuranceType = userDoc['InsuranceType'];  
 if (kDebugMode) {  
 print("Fetched insurance type dddddddddd: $insuranceType");  
 }  
 });  
 }  
 } catch (e) {  
 if (kDebugMode) {  
 print("Error fetching insurance type: $e");  
 }  
 }  
 }  
  
 @override  
 void initState() {  
 super.initState();  
 \_fetchInsuranceType(); // Fetch the insurance type on init  
 }  
  
 @override  
 void didUpdateWidget(Claimpage oldWidget) {  
 super.didUpdateWidget(oldWidget);  
 // Fetch insurance type again if the widget id has changed (user has logged out and logged in with a new id)  
 if (widget.id != oldWidget.id) {  
 \_fetchInsuranceType();  
 }  
 }  
  
  
 Future<void> \_submitClaim() async {  
 if (\_formKey.currentState!.validate()) {  
 // Check if the image is selected  
 if (\_image == null && \_imageBase64 == null) {  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(content: Text("Please upload an image")));  
 return;  
 }  
  
 // Check if the "Type of Admission" is selected  
 if (\_selectedValue == null) {  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(content: Text("Please select a type of admission")));  
 return;  
 }  
  
 try {  
 String userId = widget.id;  
  
 // Prepare data for submission  
 var claimData = {  
  
 'hospitalName': hpnamecontroller.text,  
 'hospitalNumber': hpnumbercontroller.text,  
 'hospitalIpNumber': hpipnumbercontroller.text,  
 'typeOfAdmission': \_selectedValue,  
 'firNumber': firnumbercontroller.text,  
 'billNumber': billnumbercontroller.text,  
 'amount': amountcontroller.text,  
 'image': kIsWeb ? \_imageBase64 : null,  
 'timestamp': FieldValue.*serverTimestamp*(),  
 "aprovelstatus": "pending",  
 };  
  
 await \_firestore.collection('Requests').doc(widget.id).set({  
 'patientName':patientnamecontroller.text,  
 'hospitalName': hpnamecontroller.text,  
 'hospitalNumber': hpnumbercontroller.text,  
 'hospitalIpNumber': hpipnumbercontroller.text,  
 'typeOfAdmission': \_selectedValue,  
 'firNumber': firnumbercontroller.text,  
 'billNumber': billnumbercontroller.text,  
 'amount': amountcontroller.text,  
 'image': kIsWeb ? \_imageBase64 : null,  
 'timestamp': FieldValue.*serverTimestamp*(),  
 'status':false,// Add a timestamp for reference  
 "aprovelstatus": "pending",  
 'id':widget.id,  
 });  
 // Add claim data to the user's Claim subcollection  
 await \_firestore.collection('Users').doc(userId).collection('Claim').add(claimData);  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(content: Text("Claim Submitted Successfully")));  
 } catch (e) {  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(content: Text("Error submitting claim")));  
 }  
 }  
 }  
  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
  
 backgroundColor: Colors.*white*,  
  
  
 ),  
  
 body: Container(  
 width: 1440.w,  
 height: 833.h,  
 decoration: BoxDecoration(image: DecorationImage(image: AssetImage("assets/claimbg.jpg"), fit: BoxFit.fill)),  
 child: Center(  
 child: Container(  
 width: 700.w,  
 height: 833.h,  
 color: Colors.*white*.withOpacity(.9),  
 child: Padding(  
 padding: EdgeInsets.only(left: 20.w),  
 child: SingleChildScrollView(  
 child: Form(  
 key: \_formKey,  
 child: Column(  
 children: [  
 if (insuranceType == "Full Cover") ...[  
 Column(crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 SizedBox(height: 20.h),  
 Center(  
 child: Text(  
 "Device Details ",  
 style: TextStyle(color: Colors.*black*, fontSize: 22.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
  
 Text(  
 "Engine Number : 12345566 ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 Text(  
 "Cheers Number : 67890989",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),Text(  
 "Speed :100 ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),Text(  
 "Accident Location : palakkad ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 Text(  
 "Engine Temprature : 100 ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
  
 Center(  
 child: Text(  
 "Medical Details",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 SizedBox(height: 20.h),  
 Row(  
 children: [  
 Text(  
 "patient Name : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: patientnamecontroller,  
 decoration: InputDecoration(  
 hintText: "patient name ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'patient name is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
 // Hospital Name input  
 Row(  
 children: [  
 Text(  
 "Hospital Name : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpnamecontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital name ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital name is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Hospital Number input  
 Row(  
 children: [  
 Text(  
 "Hospital Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Hospital IP Number input  
 Row(  
 children: [  
 Text(  
 "Hospital Ip Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpipnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital Ip Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital IP number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Type of Admission dropdown  
 Row(  
 children: [  
 Text(  
 "Type of Admission : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 DropdownButton<String>(  
 value: \_selectedValue,  
 hint: Text("Select"), // Display "Select" when empty  
 onChanged: (String? newValue) {  
 setState(() {  
 \_selectedValue = newValue;  
 });  
 },  
 items: <String>['Casualty', 'Emergency', 'OP']  
 .map<DropdownMenuItem<String>>((String value) {  
 return DropdownMenuItem<String>(  
 value: value,  
 child: SizedBox(  
 width: 350.w,  
 height: 50.h,  
 child: Text(value),  
 ),  
 );  
 }).toList(),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // FIR Number input  
 Row(  
 children: [  
 Text(  
 "FIR Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: firnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "FIR Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'FIR number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Bill Number input  
 Row(  
 children: [  
 Text(  
 "Bill Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: billnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Bill Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Bill number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Amount input  
 Row(  
 children: [  
 Text(  
 "Amount : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: amountcontroller,  
 decoration: InputDecoration(  
 hintText: "Amount ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Amount is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Upload Image row  
 Row(  
 children: [  
 Text(  
 "Upload Image : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 GestureDetector(  
 onTap: \_pickImage, // Open the gallery when tapped  
 child: Container(  
 width: 150.w,  
 height: 40.h,  
 decoration: ShapeDecoration(  
 color: Colors.*blue*,  
 shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20.r)),  
 ),  
 child: Center(  
 child: Text(  
 "Choose Photo",  
 style: TextStyle(color: Colors.*black*, fontSize: 15.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 ),  
 ),  
 SizedBox(width: 50.w),  
 Container(  
 width: 100.w,  
 height: 120.h,  
 color: Colors.*grey*[300],  
 child: kIsWeb  
 ? (\_imageBase64 == null  
 ? Center(child: Text("No Image"))  
 : Image.memory(base64Decode(\_imageBase64!), fit: BoxFit.cover))  
 : (\_image == null  
 ? Center(child: Text("No Image"))  
 : Image.file(\_image!, fit: BoxFit.fill)),  
 ),  
 ],  
 ),  
 SizedBox(height: 30.h),  
  
 Center(  
 child: InkWell(  
 onTap: \_submitClaim,  
 child: Container(  
 width: 150.w,  
 height: 50.h,  
 decoration: ShapeDecoration(  
 color: Colors.*blue*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(20.r),  
 ),  
 ),  
 child: Center(  
 child: Text(  
 "Submit",  
 style: TextStyle(color: Colors.*black*, fontSize: 15.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 ),  
 ),  
 ),  
 SizedBox(height: 20.h),  
 ],  
 ),  
 ] else if (insuranceType == "Third Party") ...[  
 Column(  
 children: [  
 SizedBox(height: 20.h),  
 Center(  
 child: Text(  
 "Medical Details",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 SizedBox(height: 20.h),  
 Row(  
 children: [  
 Text(  
 "patient Name : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: patientnamecontroller,  
 decoration: InputDecoration(  
 hintText: "patient name ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'patient name is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
 // Hospital Name input  
 Row(  
 children: [  
 Text(  
 "Hospital Name : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpnamecontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital name ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital name is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Hospital Number input  
 Row(  
 children: [  
 Text(  
 "Hospital Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Hospital IP Number input  
 Row(  
 children: [  
 Text(  
 "Hospital Ip Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: hpipnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Hospital Ip Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Hospital IP number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Type of Admission dropdown  
 Row(  
 children: [  
 Text(  
 "Type of Admission : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 DropdownButton<String>(  
 value: \_selectedValue,  
 hint: Text("Select"), // Display "Select" when empty  
 onChanged: (String? newValue) {  
 setState(() {  
 \_selectedValue = newValue;  
 });  
 },  
 items: <String>['Casualty', 'Emergency', 'OP']  
 .map<DropdownMenuItem<String>>((String value) {  
 return DropdownMenuItem<String>(  
 value: value,  
 child: SizedBox(  
 width: 350.w,  
 height: 50.h,  
 child: Text(value),  
 ),  
 );  
 }).toList(),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // FIR Number input  
 Row(  
 children: [  
 Text(  
 "FIR Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: firnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "FIR Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'FIR number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Bill Number input  
 Row(  
 children: [  
 Text(  
 "Bill Number : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: billnumbercontroller,  
 decoration: InputDecoration(  
 hintText: "Bill Number ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Bill number is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Amount input  
 Row(  
 children: [  
 Text(  
 "Amount : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(  
 width: 400.w,  
 child: TextFormField(  
 controller: amountcontroller,  
 decoration: InputDecoration(  
 hintText: "Amount ",  
 border: OutlineInputBorder(),  
 filled: true,  
 fillColor: Colors.*white*,  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Amount is required';  
 }  
 return null;  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20.h),  
  
 // Upload Image row  
 Row(  
 children: [  
 Text(  
 "Upload Image : ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20.sp, fontWeight: FontWeight.*bold*),  
 ),  
 GestureDetector(  
 onTap: \_pickImage, // Open the gallery when tapped  
 child: Container(  
 width: 150.w,  
 height: 40.h,  
 decoration: ShapeDecoration(  
 color: Colors.*blue*,  
 shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20.r)),  
 ),  
 child: Center(  
 child: Text(  
 "Choose Photo",  
 style: TextStyle(color: Colors.*black*, fontSize: 15.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 ),  
 ),  
 SizedBox(width: 50.w),  
 Container(  
 width: 100.w,  
 height: 120.h,  
 color: Colors.*grey*[300],  
 child: kIsWeb  
 ? (\_imageBase64 == null  
 ? Center(child: Text("No Image"))  
 : Image.memory(base64Decode(\_imageBase64!), fit: BoxFit.cover))  
 : (\_image == null  
 ? Center(child: Text("No Image"))  
 : Image.file(\_image!, fit: BoxFit.fill)),  
 ),  
 ],  
 ),  
 SizedBox(height: 30.h),  
  
 InkWell(  
 onTap: \_submitClaim,  
 child: Container(  
 width: 150.w,  
 height: 50.h,  
 decoration: ShapeDecoration(  
 color: Colors.*blue*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(20.r),  
 ),  
 ),  
 child: Center(  
 child: Text(  
 "Submit",  
 style: TextStyle(color: Colors.*black*, fontSize: 15.sp, fontWeight: FontWeight.*bold*),  
 ),  
 ),  
 ),  
 ),  
 SizedBox(height: 20.h),  
 ],  
 ),  
 ] else ...[  
 SizedBox()  
 // Maybe a loading or default state if no insuranceType is fetched yet.  
 ]  
 ],  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
}