import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter\_screenutil/flutter\_screenutil.dart';  
import 'package:insuranceadminside/Home.dart';  
  
class Login extends StatefulWidget {  
 const Login({super.key});  
  
 @override  
 State<Login> createState() => \_LoginState();  
}  
  
class \_LoginState extends State<Login> {  
 bool \_isPasswordVisible = false; // State for showing/hiding password  
 final \_emailController = TextEditingController(); // Email Controller  
 final \_passwordController = TextEditingController(); // Password Controller  
  
 final \_formKey = GlobalKey<FormState>(); // Global key for form validation  
 FirebaseAuth auth = FirebaseAuth.*instance*;  
  
 // Function to handle login  
 void \_login() async {  
 if (\_formKey.currentState!.validate()) {  
 try {  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(  
 content: Text("Logging in..."),  
 ));  
  
 // Firebase authentication login  
 UserCredential userCredential = await auth.signInWithEmailAndPassword(  
 email: \_emailController.text,  
 password: \_passwordController.text,  
 );  
  
 if (userCredential.user != null) {  
 // Successful login, navigate to the Home screen  
 Navigator.*of*(context).pushAndRemoveUntil(  
 MaterialPageRoute(builder: (\_) => Home()),  
 (route) => false  
 );  
 }  
 } on FirebaseAuthException catch (e) {  
 String errorMessage = 'An error occurred';  
  
 if (e.code == 'user-not-found') {  
 errorMessage = 'No user found for that email.';  
 } else if (e.code == 'wrong-password') {  
 errorMessage = 'Wrong password provided for that user.';  
 }  
  
 ScaffoldMessenger.*of*(context).showSnackBar(SnackBar(  
 content: Text(errorMessage),  
 ));  
 }  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Container(  
 width: 1440.w, // Full container width  
 height: 800.h, // Full container height  
 decoration: BoxDecoration(  
 image: DecorationImage(  
 image: AssetImage("assets/insu.webp"), // Background image  
 fit: BoxFit.cover, // Full-screen image  
 ),  
 ),  
 child: Padding(  
 padding: EdgeInsets.only(right: 60.w),  
 child: Align(  
 alignment: Alignment.*centerRight*,  
 child: SizedBox(  
 width: 500.w, // Set the Card width to 500 logical pixels  
 height: 700.h, // Set the height of the Card  
 child: Card(  
 color: Colors.*white*,  
 elevation: 15, // Higher elevation for more emphasis  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(20), // Rounded corners  
 ),  
 child: Padding(  
 padding: EdgeInsets.all(16.w),  
 child: Form(  
 key: \_formKey, // Assigning the form key  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Center(  
 child: Text(  
 "Login",  
 style: TextStyle(  
 color: Colors.*blue*,  
 fontWeight: FontWeight.*bold*,  
 fontSize: 30.sp,  
 ),  
 ),  
 ),  
 SizedBox(height: 50.h),  
 \_buildTextField("Email", "Enter your email", false, \_emailController),  
 SizedBox(height: 20.h),  
 \_buildTextField("Password", "\*\*\*\*\*\*\*\*\*\*\*\*", true, \_passwordController),  
 SizedBox(height: 50.h),  
 Center(  
 child: ElevatedButton(  
 onPressed: \_login,  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blueAccent*,  
 padding: EdgeInsets.symmetric(  
 horizontal: 100.w,  
 vertical: 16.h,  
 ),  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(15),  
 ),  
 ),  
 child: Text(  
 "Login",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 20.sp,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
  
 // A reusable method to create text fields with validation and password visibility toggle  
 Widget \_buildTextField(String label, String hintText, bool isPassword, TextEditingController controller) {  
 return Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(  
 label,  
 style: TextStyle(  
 color: Colors.*blue*,  
 fontSize: 20.sp,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 SizedBox(height: 20.h),  
 SizedBox(  
 width: 400.w, // Adjust the TextField width (example: 300.w)  
 child: TextFormField(  
 controller: controller,  
 obscureText: isPassword && !\_isPasswordVisible,  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return '$label is required';  
 }  
 if (label == 'Email' && !RegExp(r'^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$').hasMatch(value)) {  
 return 'Please enter a valid email';  
 }  
 return null;  
 },  
 decoration: InputDecoration(  
 hintText: hintText,  
 border: OutlineInputBorder(),  
 focusedBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*blueAccent*, width: 2.0),  
 ),  
 enabledBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*grey*, width: 1.0),  
 ),  
 suffixIcon: isPassword  
 ? IconButton(  
 icon: Icon(  
 \_isPasswordVisible ? Icons.*visibility* : Icons.*visibility\_off*,  
 color: Colors.*blue*,  
 ),  
 onPressed: () {  
 setState(() {  
 \_isPasswordVisible = !\_isPasswordVisible;  
 });  
 },  
 )  
 : null,  
 ),  
 ),  
 ),  
 ],  
 );  
 }  
}