Working codes as for now

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
main.dart

import 'package:flutter/material.dart';

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

import 'package:flutter\_gen/gen\_l10n/app\_localizations.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'firebase\_options.dart';

import 'home\_page.dart';

import 'view\_card\_page.dart';

void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);

  final prefs = await SharedPreferences.getInstance();

  final String langCode = prefs.getString('language') ?? 'en';

  final bool isLoggedIn = prefs.getBool('isLoggedIn') ?? false;

  final String userEmail = prefs.getString('userEmail') ?? '';

  runApp(MyApp(Locale(langCode), isLoggedIn, userEmail));

}

class MyApp extends StatefulWidget {

  final Locale initialLocale;

  final bool isLoggedIn;

  final String userEmail;

  const MyApp(this.initialLocale, this.isLoggedIn, this.userEmail, {super.key});

  @override

  MyAppState createState() => MyAppState();

  static MyAppState? of(BuildContext context) =>

      context.findAncestorStateOfType<MyAppState>();

}

class MyAppState extends State<MyApp> {

  late Locale \_locale;

  late bool \_isLoggedIn;

  late String \_userEmail;

  @override

  void initState() {

    super.initState();

    \_locale = widget.initialLocale;

    \_isLoggedIn = widget.isLoggedIn;

    \_userEmail = widget.userEmail;

  }

  void setLocale(Locale newLocale) async {

    final prefs = await SharedPreferences.getInstance();

    await prefs.setString('language', newLocale.languageCode);

    setState(() {

      \_locale = newLocale;

    });

  }

  void login(String email) async {

    final prefs = await SharedPreferences.getInstance();

    await prefs.setBool('isLoggedIn', true);

    await prefs.setString('userEmail', email);

    setState(() {

      \_isLoggedIn = true;

      \_userEmail = email;

    });

  }

  void logout() async {

    final prefs = await SharedPreferences.getInstance();

    await prefs.setBool('isLoggedIn', false);

    await prefs.remove('userEmail');

    setState(() {

      \_isLoggedIn = false;

      \_userEmail = '';

    });

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'Gov ID Card Generator',

      locale: \_locale,

      localizationsDelegates: const [

        AppLocalizations.delegate,

        GlobalMaterialLocalizations.delegate,

        GlobalWidgetsLocalizations.delegate,

        GlobalCupertinoLocalizations.delegate,

      ],

      supportedLocales: const [Locale('en', ''), Locale('ta', '')],

      home:

          \_isLoggedIn

              ? ViewCardPage(email: \_userEmail, onLogout: logout)

              : HomePage(onLogout: logout),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-----------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Register.dart  
import 'package:flutter/material.dart';

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

import 'dart:math';

import 'package:flutter\_gen/gen\_l10n/app\_localizations.dart';

class RegisterPage extends StatefulWidget {

  const RegisterPage({super.key});

  @override

  \_RegisterPageState createState() => \_RegisterPageState();

}

class \_RegisterPageState extends State<RegisterPage> {

  final TextEditingController \_nameController = TextEditingController();

  final TextEditingController \_dobController = TextEditingController();

  final TextEditingController \_addressController = TextEditingController();

  final TextEditingController \_emailController = TextEditingController();

  final TextEditingController \_passwordController = TextEditingController();

  final CollectionReference users = FirebaseFirestore.instance.collection(

    'users',

  );

  String \_generateRandomId() {

    final Random random = Random();

    return 'ID${random.nextInt(900000) + 100000}'; // Generates a 6-digit ID

  }

  void \_registerUser() async {

    if (\_nameController.text.isEmpty ||

        \_dobController.text.isEmpty ||

        \_addressController.text.isEmpty ||

        \_emailController.text.isEmpty ||

        \_passwordController.text.isEmpty) {

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text(AppLocalizations.of(context)!.fillAllFields)),

      );

      return;

    }

    // Validate email format

    String email = \_emailController.text.trim().toLowerCase();

    if (!email.contains('@')) {

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text(AppLocalizations.of(context)!.invalidEmail)),

      );

      return;

    }

    // Validate password strength

    if (\_passwordController.text.length < 6) {

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text(AppLocalizations.of(context)!.weakPassword)),

      );

      return;

    }

    String generatedId = \_generateRandomId();

    try {

      print("Registering user: $email");

      await users.doc(email).set({

        // 🔥 Using email as the document ID!

        'name': \_nameController.text,

        'dob': \_dobController.text,

        'address': \_addressController.text,

        'email': email,

        'password': \_passwordController.text, // Hash in real apps

        'idNumber': generatedId,

      });

      print("User registered successfully: $email");

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(

          content: Text(

            '${AppLocalizations.of(context)!.registerSuccess} ID: $generatedId',

          ),

        ),

      );

      \_nameController.clear();

      \_dobController.clear();

      \_addressController.clear();

      \_emailController.clear();

      \_passwordController.clear();

    } catch (e) {

      print("Error registering user: $e");

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text(AppLocalizations.of(context)!.registerFailed)),

      );

    }

  }

  @override

  Widget build(BuildContext context) {

    var locale = AppLocalizations.of(context)!;

    return Scaffold(

      appBar: AppBar(title: Text(locale.register)),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          children: [

            TextField(

              controller: \_nameController,

              decoration: InputDecoration(labelText: locale.fullName),

            ),

            TextField(

              controller: \_dobController,

              decoration: InputDecoration(labelText: locale.dob),

            ),

            TextField(

              controller: \_addressController,

              decoration: InputDecoration(labelText: locale.address),

            ),

            TextField(

              controller: \_emailController,

              decoration: InputDecoration(labelText: locale.email),

              keyboardType: TextInputType.emailAddress,

            ),

            TextField(

              controller: \_passwordController,

              decoration: InputDecoration(labelText: locale.password),

              obscureText: true,

            ),

            const SizedBox(height: 20),

            ElevatedButton(

              onPressed: \_registerUser,

              child: Text(locale.register),

            ),

          ],

        ),

      ),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-------------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

import 'package:flutter/material.dart';

import 'package:flutter\_gen/gen\_l10n/app\_localizations.dart';

class RegisteredDetailsPage extends StatelessWidget {

  const RegisteredDetailsPage({super.key});

  @override

  Widget build(BuildContext context) {

    var locale = AppLocalizations.of(context)!;

    return Scaffold(

      appBar: AppBar(title: Text(locale.viewUsers)),

      body: StreamBuilder<QuerySnapshot>(

        stream: FirebaseFirestore.instance.collection('users').snapshots(),

        builder: (context, snapshot) {

          if (snapshot.connectionState == ConnectionState.waiting) {

            return const Center(child: CircularProgressIndicator());

          }

          if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {

            return Center(child: Text(locale.noUsers));

          }

          return ListView(

            padding: const EdgeInsets.all(10),

            children:

                snapshot.data!.docs.map((doc) {

                  Map<String, dynamic> data =

                      doc.data() as Map<String, dynamic>;

                  return Card(

                    elevation: 3,

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(10),

                    ),

                    child: ListTile(

                      title: Text(

                        data['name'],

                        style: const TextStyle(fontWeight: FontWeight.bold),

                      ),

                      subtitle: Column(

                        crossAxisAlignment: CrossAxisAlignment.start,

                        children: [

                          Text('${locale.dob}: ${data['dob']}'),

                          Text('${locale.address}: ${data['address']}'),

                          Text(

                            'ID: ${data['idNumber']}',

                            style: const TextStyle(fontWeight: FontWeight.bold),

                          ),

                        ],

                      ),

                    ),

                  );

                }).toList(),

          );

        },

      ),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-------------------------------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Home.dart

import 'package:flutter/material.dart';

import 'package:flutter\_gen/gen\_l10n/app\_localizations.dart';

import 'register\_page.dart';

import 'login\_page.dart';

import 'main.dart';

class HomePage extends StatelessWidget {

  final VoidCallback onLogout;

  const HomePage({super.key, required this.onLogout});

  @override

  Widget build(BuildContext context) {

    var locale = AppLocalizations.of(context)!;

    return Scaffold(

      appBar: AppBar(

        title: Text(locale.appTitle),

        actions: [

          DropdownButton<Locale>(

            onChanged: (Locale? newLocale) {

              if (newLocale != null) {

                MyApp.of(context)?.setLocale(newLocale);

              }

            },

            icon: const Icon(Icons.language),

            items: const [

              DropdownMenuItem(value: Locale('en', ''), child: Text('English')),

              DropdownMenuItem(value: Locale('ta', ''), child: Text('தமிழ்')),

            ],

          ),

        ],

      ),

      body: Center(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            ElevatedButton(

              onPressed: () {

                Navigator.push(

                  context,

                  MaterialPageRoute(builder: (context) => const RegisterPage()),

                );

              },

              child: Text(locale.register),

            ),

            const SizedBox(height: 10),

            ElevatedButton(

              onPressed: () {

                Navigator.push(

                  context,

                  MaterialPageRoute(builder: (context) => const LoginPage()),

                );

              },

              child: Text(locale.login),

            ),

          ],

        ),

      ),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_--------------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Viewcard.dart

import 'package:flutter/material.dart';

import 'package:qr\_flutter/qr\_flutter.dart';

import 'package:gov\_id/database\_helper.dart';

import 'home\_page.dart';

class ViewCardPage extends StatefulWidget {

  final String email;

  final VoidCallback onLogout;

  const ViewCardPage({super.key, required this.email, required this.onLogout});

  @override

  \_ViewCardPageState createState() => \_ViewCardPageState();

}

class \_ViewCardPageState extends State<ViewCardPage> {

  final DatabaseHelper \_dbHelper = DatabaseHelper();

  Map<String, dynamic>? \_userData;

  bool \_isLoading = true;

  @override

  void initState() {

    super.initState();

    \_fetchUserData();

  }

  void \_fetchUserData() async {

    print("Fetching user data for: ${widget.email}");

    Map<String, dynamic>? userData = await \_dbHelper.getUserDetails(

      widget.email,

    );

    if (mounted) {

      setState(() {

        \_userData = userData;

        \_isLoading = false;

      });

    }

    print("Fetched user data: $\_userData");

  }

  void \_logout() {

    print("Logging out...");

    widget.onLogout();

    Navigator.pushAndRemoveUntil(

      context,

      MaterialPageRoute(

        builder: (context) => HomePage(onLogout: widget.onLogout),

      ),

      (Route<dynamic> route) => false,

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("Your ID Card"),

        actions: [

          IconButton(icon: const Icon(Icons.logout), onPressed: \_logout),

        ],

      ),

      body:

          \_isLoading

              ? const Center(child: CircularProgressIndicator())

              : \_userData == null

              ? const Center(child: Text("User data not found."))

              : Center(

                child: Card(

                  elevation: 6,

                  shape: RoundedRectangleBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                  child: Container(

                    width: 350,

                    height: 220,

                    decoration: BoxDecoration(

                      borderRadius: BorderRadius.circular(12),

                      border: Border.all(color: Colors.black, width: 1),

                      gradient: const LinearGradient(

                        colors: [

                          Color(0xFFFF9933), // Saffron

                          Colors.white,

                          Color(0xFF138808), // Green

                        ],

                        stops: [0.2, 0.5, 0.8],

                        begin: Alignment.topLeft,

                        end: Alignment.bottomRight,

                      ),

                    ),

                    child: Padding(

                      padding: const EdgeInsets.all(12),

                      child: Row(

                        mainAxisAlignment: MainAxisAlignment.spaceBetween,

                        children: [

                          // User Info Column

                          Expanded(

                            flex: 3,

                            child: Column(

                              crossAxisAlignment: CrossAxisAlignment.start,

                              mainAxisAlignment: MainAxisAlignment.center,

                              children: [

                                const Text(

                                  "Government of India",

                                  style: TextStyle(

                                    fontSize: 16,

                                    fontWeight: FontWeight.bold,

                                    color: Colors.black87,

                                  ),

                                ),

                                const SizedBox(height: 5),

                                \_buildInfoRow("Name", \_userData!['name']),

                                \_buildInfoRow("DOB", \_userData!['dob']),

                                \_buildInfoRow("Address", \_userData!['address']),

                                //\_buildInfoRow("Email", \_userData!['email']),

                                \_buildInfoRow(

                                  "Unique ID ",

                                  "IND-${\_userData!['idNumber']}",

                                  bold: true,

                                ),

                              ],

                            ),

                          ),

                          // QR Code Column

                          Expanded(

                            flex: 2,

                            child: Column(

                              mainAxisAlignment: MainAxisAlignment.center,

                              children: [

                                QrImageView(

                                  data:

                                      "${\_userData!['name']},${\_userData!['idNumber']},${\_userData!['email']}",

                                  version: QrVersions.auto,

                                  size: 70,

                                ),

                                const SizedBox(height: 5),

                                const Text(

                                  "Scan for Details",

                                  style: TextStyle(fontSize: 10),

                                ),

                              ],

                            ),

                          ),

                        ],

                      ),

                    ),

                  ),

                ),

              ),

    );

  }

  // Helper method to create rows for displaying user data

  Widget \_buildInfoRow(String label, String value, {bool bold = false}) {

    return Padding(

      padding: const EdgeInsets.symmetric(vertical: 2),

      child: Row(

        children: [

          Text(

            "$label: ",

            style: TextStyle(

              fontWeight: FontWeight.bold,

              fontSize: bold ? 14 : 12,

            ),

          ),

          Expanded(

            child: Text(

              value,

              overflow: TextOverflow.ellipsis,

              style: TextStyle(fontSize: 12),

            ),

          ),

        ],

      ),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_------------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Loginpage.dart  
import 'package:flutter/material.dart';

import 'package:gov\_id/database\_helper.dart';

import 'package:gov\_id/view\_card\_page.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'main.dart';

class LoginPage extends StatefulWidget {

  const LoginPage({super.key});

  @override

  \_LoginPageState createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

  final TextEditingController \_emailController = TextEditingController();

  final TextEditingController \_passwordController = TextEditingController();

  final DatabaseHelper \_dbHelper = DatabaseHelper();

  void \_login() async {

    String email = \_emailController.text.trim();

    String password = \_passwordController.text.trim();

    if (email.isEmpty || password.isEmpty) {

      ScaffoldMessenger.of(context).showSnackBar(

        const SnackBar(content: Text("Please fill in all fields.")),

      );

      return;

    }

    bool isValid = await \_dbHelper.loginUser(email, password);

    if (isValid) {

      final prefs = await SharedPreferences.getInstance();

      await prefs.setBool('isLoggedIn', true);

      await prefs.setString('userEmail', email);

      Navigator.pushReplacement(

        context,

        MaterialPageRoute(

          builder:

              (context) => ViewCardPage(

                email: email,

                onLogout: MyApp.of(context)!.logout,

              ),

        ),

      );

    } else {

      ScaffoldMessenger.of(context).showSnackBar(

        const SnackBar(content: Text("Invalid email or password.")),

      );

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: const Text("Login")),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          children: [

            TextField(

              controller: \_emailController,

              decoration: const InputDecoration(labelText: "Email"),

              keyboardType: TextInputType.emailAddress,

            ),

            TextField(

              controller: \_passwordController,

              decoration: const InputDecoration(labelText: "Password"),

              obscureText: true,

            ),

            const SizedBox(height: 20),

            ElevatedButton(onPressed: \_login, child: const Text("Login")),

          ],

        ),

      ),

    );

  }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-------------------------------------------------\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

class DatabaseHelper {

  final CollectionReference users = FirebaseFirestore.instance.collection(

    'users',

  );

  // Register user (Store in Firestore)

  Future<bool> registerUser(

    String name,

    String dob,

    String address,

    String email,

    String password,

    String idNumber,

  ) async {

    try {

      String formattedEmail =

          email.trim().toLowerCase(); // Trim spaces & make lowercase

      print("Registering user: $formattedEmail");

      await users.doc(formattedEmail).set({

        // Set email as document ID

        'name': name,

        'dob': dob,

        'address': address,

        'email': formattedEmail,

        'password': password, // Hash in real apps

        'idNumber': idNumber,

      });

      print("User registered successfully: $formattedEmail");

      return true;

    } catch (e) {

      print("Error registering user: $e");

      return false;

    }

  }

  // Authenticate user login

  Future<bool> loginUser(String email, String password) async {

    try {

      String formattedEmail = email.toLowerCase();

      print("Attempting login for: $formattedEmail");

      // Fetch all users to debug document IDs

      QuerySnapshot snapshot = await users.get();

      List<String> storedUsers = snapshot.docs.map((doc) => doc.id).toList();

      print("Stored users in Firestore: $storedUsers");

      // Check if the email exists

      DocumentSnapshot user = await users.doc(formattedEmail).get();

      if (user.exists) {

        print("User document found in Firestore.");

        String storedPassword = user.get('password');

        if (storedPassword == password) {

          print("Login successful!");

          return true;

        } else {

          print("Incorrect password entered.");

          return false;

        }

      } else {

        print("User does not exist in Firestore.");

        return false;

      }

    } catch (e) {

      print("Error logging in: $e");

      return false;

    }

  }

  // Get user details

  Future<Map<String, dynamic>?> getUserDetails(String email) async {

    try {

      print("Fetching details for: $email");

      DocumentSnapshot user = await users.doc(email.toLowerCase()).get();

      if (user.exists) {

        print("User data found: ${user.data()}");

        return user.data() as Map<String, dynamic>;

      } else {

        print("No user data found.");

        return null;

      }

    } catch (e) {

      print("Error fetching user details: $e");

      return null;

    }

  }

}