

Experiment	8
Aim	Implement user authentication and registration using database with flutter
Objective	Create Registration page
	Create database
	Store, fetch, delete values from database.
	User Authentication from database
Name	Abhijeet S Jadhav
UCID	2024510021
Class	FY - MCA
Batch	A
Date of	28-04-2025
Submission	

Tashnalagy	Engage average Electron
Technology used Task	Framework- Flutter, Tool used - Android Studio
	Programming- Dart
	Frogramming- Dart Firebase
	Create a basic registration page and store data n database. Fetch and also
iask	delete few entries from database. You can use either firebase database or postgresql.
Code with	main.dart:
proper label	<pre>import 'package:flutter/material.dart';</pre>
	<pre>import 'package:firebase_core/firebase_core.dart';</pre>
	<pre>import 'package:db_connection/pages/login_page.dart';</pre>
	void main() async {
	<pre>WidgetsFlutterBinding.ensureInitialized();</pre>
	await Firebase.initializeApp(
	options: const FirebaseOptions(
	<pre>apiKey: "AIzaSyCAyKeYBdDjdTdHTTRsiwWvtReNgorWQtE",</pre>
	authDomain: "fir-d6084.firebaseapp.com",
	projectId: "fir-d6084",
	storageBucket: "fir-d6084.firebasestorage.app",
	messagingSenderId: "1024401220777",
	appId:
	"1:1024401220777:web:f9be99c04faf0d503375b8",
	databaseURL:
	databaseoni.
	"https://fir-d6084-default-rtdb.firebaseio.com",
),
);



(Autonomous Institute Affiliated to University of Mumbai) Munshi Nagar, Andheri (W), Mumbai – 400 058. Department of Computer Science and Engineering

```
runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        title: 'Firebase Auth & Database Demo',
        debugShowCheckedModeBanner: false,
        theme: ThemeData(
            primarySwatch: Colors.blue,
            useMaterial3: true,
            colorScheme: ColorScheme.fromSeed(seedColor:
Colors.blue),
        ),
        home: const LoginPage(),
     );
}
```

login_page.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:db_connection/pages/register_page.dart';
import 'package:db_connection/pages/users_page.dart';

class LoginPage extends StatefulWidget {
   const LoginPage({super.key});

   @override
   State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
   final TextEditingController emailController =
   TextEditingController();
   final TextEditingController passwordController =
```



```
TextEditingController();
  Future<void> login() async {
    if (!mounted) return;
FirebaseAuth.instance.signInWithEmailAndPassword(
        email: emailController.text.trim(),
       password: passwordController.text,
      );
      emailController.clear();
      passwordController.clear();
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('Login Successful!'),
          backgroundColor: Colors.green,
      if (!mounted) return;
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const
UsersPage()),
      );
    } on FirebaseAuthException catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('Login Failed: ${e.message ??
e.code}'),
          backgroundColor: Colors.red,
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('An unexpected error occurred:
$e'),
          backgroundColor: Colors.red,
      );
```



```
Future<void> loginWithGoogle() async {
      final GoogleSignInAccount? googleUser = await
GoogleSignIn().signIn();
      if (googleUser == null) return; // Cancelled
      final GoogleSignInAuthentication googleAuth =
          await googleUser.authentication;
      final credential = GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
FirebaseAuth.instance.signInWithCredential(credential);
      if (!mounted) return;
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const
UsersPage()),
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.red,
        ),
 void dispose() {
    emailController.dispose();
    passwordController.dispose();
    super.dispose();
```



```
Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      body: SafeArea(
        child: Column (
          children: [
            Container (
              width: double.infinity,
              padding: const
EdgeInsets.symmetric(vertical: 30, horizontal: 20),
              decoration: BoxDecoration(
                color: const Color.fromARGB(255, 68, 58,
L83),
                borderRadius: const BorderRadius.only(
                  bottomLeft: Radius.circular(30),
                  bottomRight: Radius.circular(30),
                ),
              ),
              child: const Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  Text(
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 28,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                  SizedBox(height: 5),
```



```
Expanded (
              child: Center(
                child: SingleChildScrollView(
                  padding: const
EdgeInsets.symmetric(horizontal: 24),
                  child: Container(
                    padding: const EdgeInsets.all(24),
                    decoration: BoxDecoration(
                       color: Colors.white,
                      borderRadius:
BorderRadius.circular(20),
                      boxShadow: const [
                         BoxShadow (
                           color: Colors.black12,
                          blurRadius: 12,
                          offset: Offset(0, 6),
                        ),
                    ),
                    child: Column (
                      children: [
                         TextField(
                           controller: emailController,
                           decoration: InputDecoration(
                             labelText: 'Email',
                             prefixIcon: const
Icon(Icons.email_outlined),
                             border: OutlineInputBorder(
                               borderRadius:
BorderRadius.circular(14),
                             ),
                           ),
                           keyboardType:
TextInputType.emailAddress,
                         const SizedBox(height: 18),
                         TextField(
                           controller: passwordController,
                           obscureText: true,
                           decoration: InputDecoration(
                             labelText: 'Password',
```



```
prefixIcon: const
Icon(Icons.lock outline),
                             border: OutlineInputBorder(
                               borderRadius:
BorderRadius.circular(14),
                             ),
                           ),
                         ),
                         const SizedBox(height: 25),
                         SizedBox(
                           width: double.infinity,
                           height: 50,
                           child: ElevatedButton(
                             onPressed: login,
                             style:
ElevatedButton.styleFrom(
                               elevation: 3,
                               backgroundColor: const
Color.fromARGB(255, 68, 58, 183),
                               shape:
RoundedRectangleBorder(
                                 borderRadius:
BorderRadius.circular(14),
                               ),
                             child: const Text(
                               style: TextStyle(fontSize:
8, color: Colors.white),
                         ),
                         const SizedBox(height: 18),
                         const Divider(),
                         const SizedBox(height: 16),
                         SizedBox(
                           width: double.infinity,
                           height: 50,
                           child: OutlinedButton.icon(
                             onPressed: loginWithGoogle,
                             icon: Image.asset(
                               height: 24,
```



```
label: const Text(
                               style: TextStyle(fontSize:
L6),
                             style:
OutlinedButton.styleFrom(
                               shape:
RoundedRectangleBorder(
                                 borderRadius:
BorderRadius.circular(14),
                               ),
                           ),
                         ),
                         const SizedBox(height: 20),
                         TextButton(
                           onPressed: () {
                             Navigator.push(
                               MaterialPageRoute(builder:
(context) => const RegisterPage()),
                             );
                           child: const Text(
Register here',
                             style: TextStyle(color:
Color.fromARGB(255, 68, 58, 183)),
                           ),
                         ),
                    ),
            const Padding(
              padding: EdgeInsets.only(bottom: 12.0),
```



(Autonomous Institute Affiliated to University of Mumbai) Munshi Nagar, Andheri (W), Mumbai – 400 058. Department of Computer Science and Engineering

```
child: Text(
    '© 2025 MP Lab 8 - All Rights Reserved',
    style: TextStyle(color: Colors.grey),
    ),
    ),
    ),
    ),
    ),
}
```

register_page.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase auth/firebase auth.dart';
import
class RegisterPage extends StatefulWidget {
 const RegisterPage({super.key});
 State<RegisterPage> createState() =>
RegisterPageState();
class RegisterPageState extends State<RegisterPage> {
  final TextEditingController nameController =
TextEditingController();
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController passwordController =
TextEditingController();
  final DatabaseReference databaseRef =
FirebaseDatabase.instance.ref("users");
  Future<void> register() async {
    if (!mounted) return;
      final credential = await
```



```
FirebaseAuth.instance.createUserWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text,
      );
      if (!mounted) return;
      if (credential.user != null) {
databaseRef.child(credential.user!.uid).set({
          'name': nameController.text.trim(),
          'email': emailController.text.trim(),
        });
        nameController.clear();
        emailController.clear();
        passwordController.clear();
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(
            content: Text('User Registered
Successfully!'),
            backgroundColor: Colors.green,
          ),
        );
        Navigator.pop(context);
    } on FirebaseAuthException catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('Registration Failed: ${e.message
 ? e.code}'),
          backgroundColor: Colors.red,
        ),
      );
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('An unexpected error occurred:
se<mark>'</mark>),
          backgroundColor: Colors.red,
```



```
void dispose() {
    nameController.dispose();
    emailController.dispose();
    passwordController.dispose();
    super.dispose();
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      body: SafeArea(
        child: Column (
          children: [
            Container (
              width: double.infinity,
              padding: const
EdgeInsets.symmetric(vertical: 30, horizontal: 20),
              decoration: BoxDecoration(
                color: const Color.fromARGB(255, 68, 58,
.83),
                borderRadius: const BorderRadius.only(
                  bottomLeft: Radius.circular(30),
                  bottomRight: Radius.circular(30),
                ),
              child: const Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  Text(
                    style: TextStyle(
                      color: Colors.white,
                      fontWeight: FontWeight.bold,
                    ),
```



```
SizedBox(height: 5),
              ),
            Expanded (
              child: Center(
                child: SingleChildScrollView(
                  padding: const
EdgeInsets.symmetric(horizontal: 24),
                  child: Container(
                    padding: const EdgeInsets.all(24),
                    decoration: BoxDecoration(
                      color: Colors.white,
                      borderRadius:
BorderRadius.circular(20),
                      boxShadow: const [
                        BoxShadow(
                          color: Colors.black12,
                          blurRadius: 12,
                          offset: Offset(0, 6),
                        ),
                    ),
                    child: Column(
                      children: [
                        TextField(
                           controller: nameController,
                          decoration: InputDecoration(
                             labelText: 'Name',
                            prefixIcon: const
Icon(Icons.person outline),
                             border: OutlineInputBorder(
```



```
borderRadius:
BorderRadius.circular(14),
                            ),
                           ),
                          keyboardType:
TextInputType.name,
                        const SizedBox(height: 18),
                        TextField(
                          controller: emailController,
                          decoration: InputDecoration(
                             labelText: 'Email',
                            prefixIcon: const
Icon(Icons.email outlined),
                            border: OutlineInputBorder(
                              borderRadius:
BorderRadius.circular(14),
                          ),
                           keyboardType:
TextInputType.emailAddress,
                        const SizedBox(height: 18),
                        TextField(
                          controller: passwordController,
                          obscureText: true,
                          decoration: InputDecoration(
                             labelText: 'Password (min. 6
chars)',
                            prefixIcon: const
Icon(Icons.lock outline),
                            border: OutlineInputBorder(
                              borderRadius:
BorderRadius.circular(14),
                             ),
                          ),
                        const SizedBox(height: 25),
                        SizedBox(
                          width: double.infinity,
                          height: 50,
                          child: ElevatedButton(
                            onPressed: register,
```



```
style:
ElevatedButton.styleFrom(
                               elevation: 3,
                               backgroundColor: const
Color.fromARGB(255, 68, 58, 183),
                               shape:
RoundedRectangleBorder(
                                 borderRadius:
BorderRadius.circular(14),
                               ),
                             child: const Text(
                               style: TextStyle(fontSize:
18, color: Colors.white),
                           ),
                         ),
                         const SizedBox(height: 20),
                         TextButton(
                           onPressed: () =>
Navigator.pop(context),
                           child: const Text(
Login',
                             style: TextStyle(color:
Color.fromARGB(255, 68, 58, 183)),
                         ),
                    ),
                ),
            ),
            const Padding(
              padding: EdgeInsets.only(bottom: 12.0),
              child: Text(
                style: TextStyle(color: Colors.grey),
              ),
```



```
),
],
),
),
);
}
```

```
firebase options.dart:
import 'package:firebase_core/firebase_core.dart' show
FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
      case TargetPlatform.macOS:
        return macos;
      case TargetPlatform.windows:
        return windows;
      case TargetPlatform.linux:
        throw UnsupportedError(
```

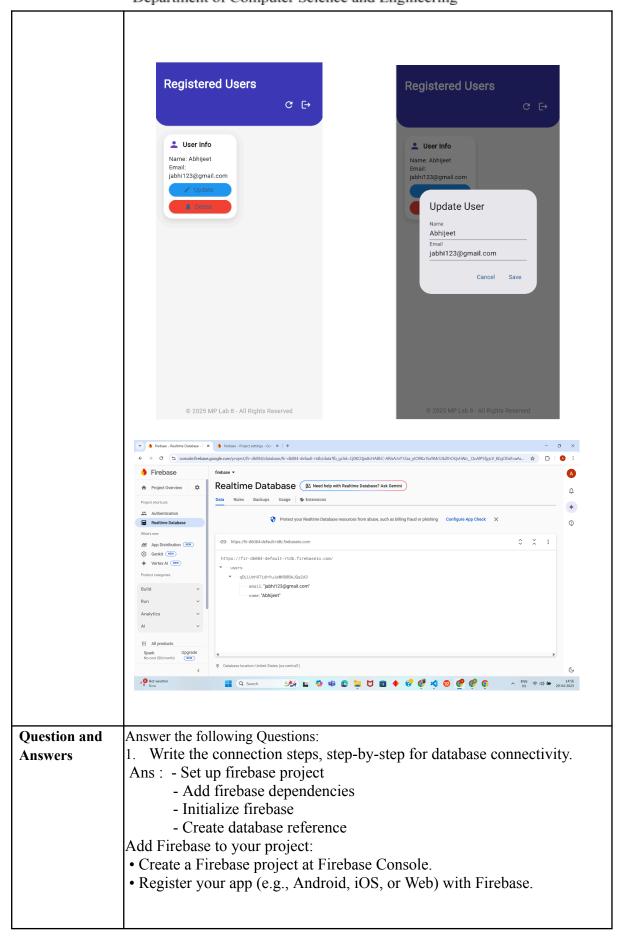


```
FlutterFire CLI again.',
        );
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for
this platform.',
  static const FirebaseOptions web = FirebaseOptions(
    apiKey: 'AIzaSyCcuJNqetq9CbZXAGEBx-N-J0rSHXkBnBI',
    appId: '1:962759037664:web:82138edafa7546a4887685',
    messagingSenderId: '962759037664',
    projectId: 'flutter-database-project-6a2a5',
    authDomain:
 flutter-database-project-6a2a5.firebaseapp.com',
    storageBucket:
    measurementId: 'G-W3714FHNG9',
  );
  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyDbGCQvIvn84nm7Fbd4HD3xgiHju9vYt8c',
    appId:
    messagingSenderId: '962759037664',
    projectId: 'flutter-database-project-6a2a5',
    storageBucket:
  static const FirebaseOptions ios = FirebaseOptions(
    apiKey: 'AIzaSyAtTG6keOvCt9 K7BXrH-9dVgplIbccNYc',
    appId: '1:962759037664:ios:230f6a4957a9ed11887685',
    messagingSenderId: '962759037664',
    projectId: 'flutter-database-project-6a2a5',
    storageBucket:
    iosBundleId: 'com.example.dbConnection',
  );
```



```
static const FirebaseOptions macos = FirebaseOptions(
                    apiKey: 'AIzaSyAtTG6keOvCt9 K7BXrH-9dVgplIbccNYc',
                    appId: '1:962759037664:ios:230f6a4957a9ed11887685',
                    messagingSenderId: '962759037664',
                    projectId: 'flutter-database-project-6a2a5',
                    storageBucket:
                    iosBundleId: 'com.example.dbConnection',
                  );
                  static const FirebaseOptions windows = FirebaseOptions(
                    apiKey: 'AIzaSyCcuJNqetq9CbZXAGEBx-N-J0rSHXkBnBI',
                    appId: '1:962759037664:web:cbd45130ff62dedb887685',
                    messagingSenderId: '962759037664',
                    projectId: 'flutter-database-project-6a2a5',
                    authDomain:
                     storageBucket:
                    measurementId: 'G-MKB10XTLV2',
                  );
Screenshots
                    User Management
                                                        Register here
                      2 Name
                      Password
                                                          Login
                                                           Password (min. 6 chars)
                                                                 Register
                                                             Already have an account? Login
                       Don't have an account? Register here
                       © 2025 MP Lab 8 - All Rights Reserved
                                                           © 2025 MP Lab 8 - All Rights Reserved
```







	Add Firebase SDK to your app: • For Flutter, add Firebase packages in pubspec.yaml Initialize Firebase in
	your app:
	• Call Firebase.initializeApp() in your main.dart before running the app
	void main() async { WidgetsFlutterBinding.ensureInitialized(); await Firebase.initializeApp(); runApp(MyApp()); } Set up Authentication (if needed).
	Connect to Firestore Database: • Use FirebaseFirestore instance to access collections/documents FirebaseFirestore.instance.collection('collectionName');
Conclusion	In this session, I learned, How to connect Flutter apps with Firebase for authentication and database storage. How to initialize and use Firestore as a cloud-hosted NoSQL database. How to perform CRUD operations (Create, Read, Update, Delete) easily using Firestore collections and documents.