Flutter Firebase Firestore: Building Real-Time Database Apps

Subtitle: Learn how to connect Flutter apps to Firebase Firestore for real-time data storage and retrieval.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app

Introduction

Firestore is a **cloud-hosted, NoSQL database** that updates data in real-time. Integrating it with Flutter allows you to **build dynamic apps** where users see live updates without refreshing.

Step 1: Add Firebase & Firestore Dependencies

```
dependencies:
  cloud_firestore: ^4.9.0
  firebase_core: ^3.27.0
```

Run:

```
flutter pub get
```

Import in Dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_core/firebase_core.dart';
```

Step 2: Initialize Firebase

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

Step 3: Adding Data to Firestore

```
FirebaseFirestore.instance.collection('users').add({
  'name': 'Hafsa',
  'email': 'hafsa@example.com',
});
```

Exercise: Add multiple user documents with different names and emails.

Step 4: Reading Data (Real-Time Stream)

```
StreamBuilder(
  stream: FirebaseFirestore.instance.collection('users').snapshots(),
  builder: (context, snapshot) {
    if (!snapshot.hasData) return CircularProgressIndicator();
    final users = snapshot.data!.docs;
    return ListView.builder(
      itemCount: users.length,
      itemBuilder: (context, index) {
        return ListTile(
          title: Text(users[index]['name']),
          subtitle: Text(users[index]['email']),
        );
      },
    );
  },
)
```

Exercise: Display all users in a scrolling list that updates in real-time.

Step 5: Updating & Deleting Data

```
// Update
FirebaseFirestore.instance.collection('users').doc(docId).update({'name': 'New N

// Delete
FirebaseFirestore.instance.collection('users').doc(docId).delete();
```

Exercise: Add buttons to update and delete user documents.

Step 6: Mini Project

Build a **Flutter Firestore CRUD App**:

Build a Flutter Firestore CRUD App!

- Add new users with a form
- Display users in a real-time list
- Update and delete users
- Handle loading and error states

Key Takeaways

- Firestore provides **real-time, cloud-based data storage**
- Use `add`, `update`, `delete` for CRUD operations
- StreamBuilder shows real-time updates automatically
- Firestore integration is essential for **dynamic, scalable apps**

Skipping Firestore integration makes your app **static and unable to sync data**, which is a critical limitation for real-world apps.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app
