Flutter Firebase Form App - Practical Exam Guide

Project Goal

Build a Flutter app with a form that:

- Stores data in Firebase Firestore

- Retrieves data from Firestore using queries

Step 1: Set Up the Project

1.1 Create a Flutter Project

```
flutter create firebase_form_app
cd firebase_form_app
```

1.2 Add Dependencies in pubspec.yaml

```
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^2.24.0
  cloud_firestore: ^4.8.4
```

Then run:

```
flutter pub get
```

Step 2: Set Up Firebase

2.1 Go to Firebase Console

- Click Add project

- Follow steps, no need for Google Analytics

2.2 Add Android App

- Register with your app s package name (e.g., com.example.firebase_form_app)

- Download google-services.json and put it in:

  android/app/google-services.json

2.3 Configure Android

In android/build.gradle, add:

```
classpath 'com.google.gms:google-services:4.3.15'
```

In android/app/build.gradle, at the bottom:

```
apply plugin: 'com.google.gms.google-services'
```

Also ensure:

```
minSdkVersion 21
```

Step 3: Initialize Firebase in App

In main.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'form_page.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Firebase Form',
      home: FormPage(),
      debugShowCheckedModeBanner: false,
    );
  }
}
```

Step 4: Create Form UI and Firestore Logic

Create a file called form_page.dart:

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class FormPage extends StatefulWidget {
  @override
  _FormPageState createState() => _FormPageState();
}
```

```
class _FormPageState extends State<FormPage> {
  final _formKey = GlobalKey<FormState>();
  final nameController = TextEditingController();
  final ageController = TextEditingController();

  final CollectionReference users = FirebaseFirestore.instance.collection('users');

  void addUser(String name, int age) {
    users.add({'name': name, 'age': age});
  }

  void queryUsersByAge(int age) {
    users.where('age', isEqualTo: age).get().then((QuerySnapshot snapshot) {
      for (var doc in snapshot.docs) {
        print('Name: ${doc['name']}, Age: ${doc['age']}');
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Firebase Form')),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
child: Column(
  children: [
    TextFormField(
      controller: nameController,
      decoration: InputDecoration(labelText: 'Enter Name'),
      validator: (value) => value!.isEmpty ? 'Enter a name' : null,
    ),
    TextFormField(
      controller: ageController,
      decoration: InputDecoration(labelText: 'Enter Age'),
      keyboardType: TextInputType.number,
      validator: (value) => value!.isEmpty ? 'Enter age' : null,
    ),
    SizedBox(height: 20),
    ElevatedButton(
      onPressed: () {
        if (_formKey.currentState!.validate()) {
          final name = nameController.text;
          final age = int.parse(ageController.text);
          addUser(name, age);
          ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('User added!')));
        }
      },
      child: Text('Submit'),
    ),
    ElevatedButton(
      onPressed: () {
```

```
                  final age = int.tryParse(ageController.text);
                  if (age != null) {
                    queryUsersByAge(age);
                  }
                },
                child: Text('Query Users by Age'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Step 5: Run the App

```
flutter run
```

Try:

- Submitting a name and age
- Querying users by age in Firestore

Firebase Firestore Structure

Collection: users

Each document:

```
{
  "name": "Alice",
  "age": 21

}
```

Bonus: Display Results in App

You can show queried users in a list using ListView or StreamBuilder.

You're Ready!

Good luck on your practical!