

Input:

**main.dart**

```
import 'package:firebase_core/firebase_core.dart';  
import 'package:flutter/material.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
import 'package:note_list_screen.dart';  
import 'package:note_provider.dart';  
import 'package:provider/provider.dart';  
import './firebase_options.dart';
```

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  runApp(  
    ChangeNotifierProvider(  
      create: (context) => NoteProvider(),  
      child: MyApp(),  
    ),  
  );  
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {
```

```
return MaterialApp(  
  title: 'Note Taking App',  
  theme: ThemeData(  
    primarySwatch: Colors.blue,  
    visualDensity: VisualDensity.adaptivePlatformDensity,  
  ),  
  home: NoteListScreen(),  
);  
}  
}
```

#### **add\_note\_screen.dart**

```
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'note_provider.dart';  
  
class AddNoteScreen extends StatelessWidget {  
  final TextEditingController _titleController = TextEditingController();  
  final TextEditingController _contentController = TextEditingController();  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Add Note'),  

```

```
),  
body: Padding(  
  padding: const EdgeInsets.all(16.0),  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      TextField(  
        controller: _titleController,  
        decoration: InputDecoration(labelText: 'Title'),  
      ),  
      SizedBox(height: 10),  
      TextField(  
        controller: _contentController,  
        decoration: InputDecoration(labelText: 'Content'),  
        maxLines: null,  
      ),  
      SizedBox(height: 20),  
      ElevatedButton(  
        onPressed: () async {  
          await _addNote(context);  
        },  
        child: Text('Add'),  
      ),  
    ],  
  ),  
),
```

```

    );
}

Future<void> _addNote(BuildContext context) async {
  String title = _titleController.text.trim();
  String content = _contentController.text.trim();

  if (title.isNotEmpty && content.isNotEmpty) {
    await Provider.of<NoteProvider>(context, listen: false)
      .addNote(title, content);
    Navigator.pop(context);
  } else {
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Error'),
        content: Text('Please enter both title and content.'),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: Text('OK'),
          ),
        ],
      ),
    );
  }
}

```

```
}}
```

### **edit\_note\_screen.dart**

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

```
class EditNoteScreen extends StatelessWidget {  
  final QueryDocumentSnapshot note;
```

```
  EditNoteScreen(this.note);
```

```
  @override
```

```
  Widget build(BuildContext context) {  
    final TextEditingController _titleController =  
      TextEditingController(text: note['title']);  
    final TextEditingController _contentController =  
      TextEditingController(text: note['content']);
```

```
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Edit Note'),  
      ),  
      body: Padding(  
        padding: const EdgeInsets.all(16.0),  
        child: Column(  

```

```
crossAxisAlignment: CrossAxisAlignment.start,
children: [
  TextField(
    controller: _titleController,
    decoration: InputDecoration(labelText: 'Title'),
  ),
  SizedBox(height: 10),
  TextField(
    controller: _contentController,
    decoration: InputDecoration(labelText: 'Content'),
    maxLines: null,
  ),
  SizedBox(height: 20),
  ElevatedButton(
    onPressed: () async {
      await _updateNote(
        context, _titleController.text, _contentController.text);
      Navigator.pop(context, {
        'title': _titleController.text,
        'content': _contentController.text,
      });
    },
    child: Text('Save'),
  ),
],
),
```

```
    ),  
    );  
}
```

```
Future<void> _updateNote(  
    BuildContext context, String title, String content) async {  
    await FirebaseFirestore.instance.collection('notes').doc(note.id).update({  
        'title': title,  
        'content': content,  
        'timestamp': DateTime.now(),  
    });  
}  
}
```

### **note\_detail\_screen.dart**

```
import 'package:flutter/material.dart';  
import 'package:not/edit_note_screen.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
  
class NoteDetailScreen extends StatefulWidget {  
    final QueryDocumentSnapshot note;  
  
    NoteDetailScreen(this.note);  
  
    @override
```

```
_NoteDetailScreenState createState() => _NoteDetailScreenState();  
}
```

```
class _NoteDetailScreenState extends State<NoteDetailScreen> {  
  String title = "";  
  String content = "";
```

```
  @override  
  void initState() {  
    super.initState();  
    title = widget.note['title'];  
    content = widget.note['content'];  
  }
```

```
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Note Details'),  
        actions: [  
          IconButton(  
            icon: Icon(Icons.delete),  
            onPressed: () => _confirmDelete(context),  
          ),  
        ],  
      ),
```



```
body: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        title,
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(height: 10),
      Text(
        content,
        style: TextStyle(fontSize: 18),
      ),
    ],
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: () async {
      final updatedNoteData = await Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => EditNoteScreen(widget.note),
        ),
      );
    },
  ),
),
```

```

    ),
  );

  if (updatedNoteData != null) {
    setState(() {
      title = updatedNoteData['title'];
      content = updatedNoteData['content'];
    });
  }
},
child: Icon(Icons.edit),
),
);
}

```

```

Future<void> _confirmDelete(BuildContext context) async {
  final confirmDelete = await showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text('Confirm Delete'),
      content: Text('Are you sure you want to delete this note?'),
      actions: [
        TextButton(
          onPressed: () => Navigator.pop(context, false),
          child: Text('Cancel'),
        ),
      ],
    ),
  );
}

```

```

        TextButton(
          onPressed: () => Navigator.pop(context, true),
          child: Text('Delete'),
        ),
      ],
    ),
  );

  if (confirmDelete == true) {
    await _deleteNote();
    Navigator.pop(context); // Navigate back after deletion
  }
}

```

```

Future<void> _deleteNote() async {
  await FirebaseFirestore.instance
    .collection('notes')
    .doc(widget.note.id)
    .delete();
}
}

```

### **note\_list\_screen.dart**

```

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

```

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'note_detail_screen.dart';
import 'add_note_screen.dart';
import 'note_provider.dart';

class NoteListScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Notes'),
      ),
      body: StreamBuilder(
        stream: FirebaseFirestore.instance.collection('notes').snapshots(),
        builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return Center(child: CircularProgressIndicator());
          }

          final notes = snapshot.data?.docs ?? [];

          if (notes.isEmpty) {
            return Center(
              child: Text('No notes available'),
            );
          }
        }
      )
    );
  }
}
```

```
return ListView.builder(  
  itemCount: notes.length,  
  itemBuilder: (context, index) {  
    final note = notes[index];  
    return ListTile(  
      title: Text(note['title']),  
      subtitle: Text(note['content']),  
      trailing: IconButton(  
        icon: Icon(Icons.delete),  
        onPressed: () => _confirmDelete(context, note),  
      ),  
      onTap: () {  
        Navigator.push(  
          context,  
          MaterialPageRoute(  
            builder: (context) => NoteDetailScreen(note),  
          ),  
        );  
      },  
    );  
  },  
);  
  
floatingActionButton: FloatingActionButton(  

```

```

onPressed: () {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => AddNoteScreen(),
    ),
  );
},
child: Icon(Icons.add),
),
);
}

```

```

Future<void> _confirmDelete(
  BuildContext context, QueryDocumentSnapshot note) async {
  final confirmDelete = await showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text('Confirm Delete'),
      content: Text('Are you sure you want to delete this note?'),
      actions: [
        TextButton(
          onPressed: () => Navigator.pop(context, false),
          child: Text('Cancel'),
        ),
        TextButton(

```

```
        onPressed: () => Navigator.pop(context, true),  
        child: Text('Delete'),  
      ),  
    ],  
  ),  
);
```

```
if (confirmDelete == true) {  
  await _deleteNote(note);  
}  
}
```

```
Future<void> _deleteNote(QueryDocumentSnapshot note) async {  
  await FirebaseFirestore.instance.collection('notes').doc(note.id).delete();  
}  
}
```

### **note\_provider.dart**

```
import 'package:flutter/material.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
class NoteProvider extends ChangeNotifier {  
  List<QueryDocumentSnapshot> _notes = [];  
  List<QueryDocumentSnapshot> get notes => _notes;  
  
  void setNotes(List<QueryDocumentSnapshot> notes) {
```

```

    _notes = notes;
    notifyListeners();
}

Future<void> fetchNotes() async {
    final snapshot = await FirebaseFirestore.instance.collection('notes').get();
    _notes = snapshot.docs;
    notifyListeners();
}

Future<void> updateNote(
    QueryDocumentSnapshot note, String title, String content) async {
    await FirebaseFirestore.instance.collection('notes').doc(note.id).update({
        'title': title,
        'content': content,
        'timestamp': DateTime.now(),
    });
    await fetchNotes(); // Fetch updated notes after the update
}

Future<void> addNote(String title, String content) async {
    await FirebaseFirestore.instance.collection('notes').add({
        'title': title,
        'content': content,
        'timestamp': DateTime.now(),
    });
    await fetchNotes(); // Fetch updated notes after adding a new note
}
}

```