

Notes Taking App — Project Report

1. Real World Problem Identification

1.1 Problem Overview

- Many users need a **simple, fast, offline-capable notes app** for quick and secure note-taking.
 - Users require **CRUD operations** (Create, Read, Update, Delete) with **minimal latency** and **no network dependency**.
 - A **lightweight, maintainable codebase** compatible with **mobile (Android/iOS)** and **web** platforms is desirable.
-

2. Proposed Solution

2.1 Overview

The proposed solution is a **Flutter-based Notes App** that stores data locally using **Hive**, a lightweight key-value database.

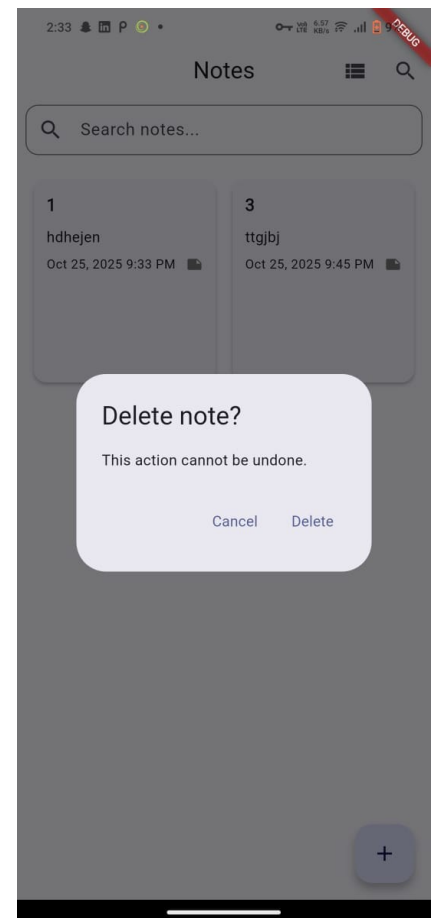
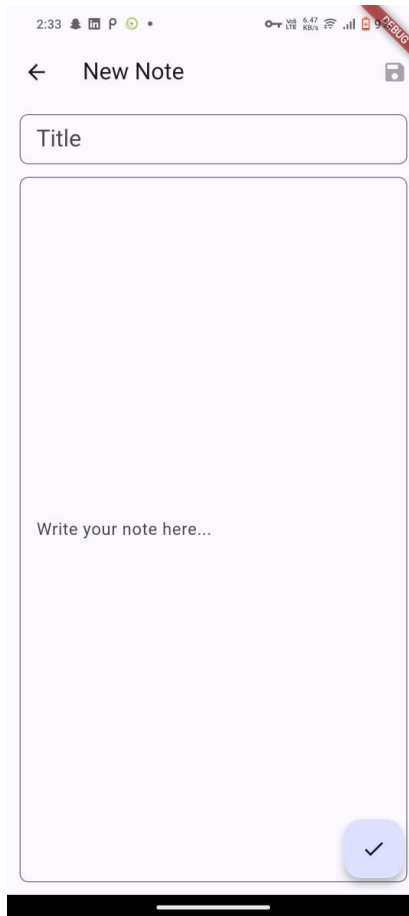
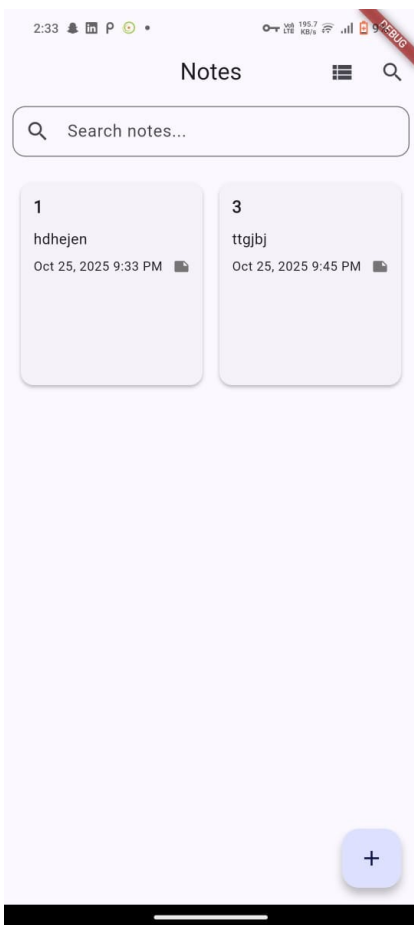
2.2 Core Functionalities

1. **Create / Edit Notes**
 - Dedicated editor screen for adding and modifying notes.
2. **Read / List Notes**
 - Home screen displays notes in both **list** and **grid** views.
3. **Delete Notes**
 - Notes can be deleted through **swipe** or **long-press** actions with confirmation.

2.3 Implementation Highlights

- Uses a **manual TypeAdapter** for `NoteModel` to avoid build-time code generation.
 - Employs **ValueListenableBuilder** with Hive's reactive listeners for **instant UI updates**.
-

3. Responsive User Interfaces



4. Data Storage

4.1 Database

- **Hive** — a local, file-based key-value store optimized for Flutter.

4.2 Justification

1. **Offline-first:** Fully local storage, works without internet.
2. **Performance:** Fast read/write operations suitable for small datasets like notes.
3. **Simplicity:** No SQL schema or setup required.
4. **Low Footprint:** Lightweight dependency with minimal startup overhead.
5. **Reactive Integration:** `Hive.box().listenable()` works seamlessly with Flutter's reactive UI system.

4.3 Implementation Details

- **Box name:** notesBox
 - **Model fields:** title, content, createdAt, lastModified
 - **Adapter file:** lib/models/note_model_adapter.dart
 - **Adapter type:** Manual TypeAdapter (avoids build_runner dependency)
-

5. APIs / Packages / Plug-ins (with Justifications)

Package	Purpose	Justification
flutter	UI framework	Core cross-platform UI toolkit
hive	Local storage	Fast, schema-light database
hive_flutter	Flutter integration	Simplifies Hive setup and reactive UI
intl	Date/time formatting	Human-readable timestamps for notes

6. Issues and Bugs Encountered (and Resolutions)

6.1 Issue 1 — List View Not Displaying Notes

Symptoms:

Grid view worked, but list view showed no notes.

Probable Causes:

1. Incorrect data source or conversion of `box.values`.
2. `ListView` improperly nested (missing `Expanded` or `shrinkWrap`).
3. UI rendering before Hive box initialization.

Fix:

Use a `ValueListenableBuilder<Box<NoteModel>>` to reactively build the list from Hive values.

```
ValueListenableBuilder<Box<NoteModel>>(  
  valueListenable: Hive.box<NoteModel>('notesBox').listenable(),  
  builder: (context, box, _) {  
    final notes = box.values.toList().cast<NoteModel>();  
    if (notes.isEmpty) {  
      return Center(child: Text('No notes yet'));  
    }  
    return ListView.builder(  
      itemCount: notes.length,  
      itemBuilder: (context, index) {  
        final note = notes[index];  
        return NoteTile(note: note);  
      },  
    );  
  },  
)
```

Also, ensure Hive is initialized **before** app launch:

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Hive.initFlutter();  
  Hive.registerAdapter(NoteModelAdapter());  
  await Hive.openBox<NoteModel>('notesBox');  
  runApp(const MyApp());  
}
```

6.2 Issue 2 — “BuildContext Used Across Async Gaps” Warning

Cause:

Using `BuildContext` after an `await` in async methods.

Fix:

Check `if (!mounted) return;` after every `await` before using context.

6.3 Other Minor Issues

- **Date/time serialization:** Ensured consistent `DateTime` read/write in adapter.
 - **Dismissible key collisions:** Resolved by using note key or UUID instead of list index.
-

7. Deliverables

7.1 Project Deliverables

1. Source code (Flutter project repository).
 2. Project report (this document).
-

8. Summary

- The app successfully implements an **offline-first notes system** using **Flutter + Hive**.
- The main runtime issue (empty list view) was resolved by correctly binding Hive data to reactive UI builders.
- The project demonstrates **clean architecture**, **local persistence**, and **responsive UI design** across platforms.
- Next step: finalize and attach screenshots, rebuild the project after applying the fixes, and export the Word report.