**Project Submission Report by: 22SW043 & 22SW122**

**📝 Task Manager App Report**

**1. Real World Problem Identification**

People frequently find it difficult to remember their daily responsibilities, due dates, and priorities in today's hectic world.   
Many people rely on random notes, reminders, or even their own memory, which results in poor time management, missed deadlines, and lower productivity.  
Without getting tangled down in too complicated apps, professionals, students, and even stay-at-home moms need a straightforward and dependable method to efficiently manage their daily tasks.

**2. Proposed Solution**

To solve this issue, we developed a **Task Manager App** using **Flutter**.  
The app allows users to:

* ✏️ Add new tasks with descriptions and due dates
* 🗓️ Edit or delete existing tasks
* ✅ Mark tasks as completed
* 💾 Store all tasks locally using Hive database
* 📱 Maintain responsive UI across mobile, tablet, and web

This app ensures that users can manage their time and workload efficiently, stay organized, and never forget important deadlines.**3. Responsive User Interfaces**

The UI is designed to be minimal, clean, and easy to understand.  
It works smoothly on Android Emulator, Chrome (Web), and different screen sizes.

**Screenshots**

**A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A white rectangular object with a black border

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A white rectangular object with a black border

AI-generated content may be incorrect.**

**4. Data Storage**

We used **Hive Database** for storing user tasks locally.

**Justification for Using Hive**

* ⚡ **Fast performance:** Hive is a lightweight, NoSQL key-value database written in Dart and optimized for Flutter apps.
* 📶 **Offline support:** Works perfectly without internet, making it ideal for a personal task manager.
* 🧩 **Simple integration:** No need for complex SQL queries — easy CRUD operations using boxes.
* 💾 **Persistent storage:** Data remains saved even after closing the app.
* 👌 **Provider integration:** Works well with the Provider state-management approach for updating the UI in real time.

**Why not SharedPreferences and Firebase**

* **SharedPreferences:** Only suitable for storing small key-value data like settings or login info. It cannot efficiently store complex structured data such as task lists with titles, descriptions, and due dates.
* **Firebase:** Requires internet and authentication setup. Our app is designed to work fully offline, so using a cloud database would add unnecessary complexity and dependency.

**5. APIs / Packages / Plug-ins Used**

| **Package** | **Purpose** | **Justification** |
| --- | --- | --- |
| **provider** | State management | Keeps UI updated automatically when data changes |
| **hive** | Local database | Lightweight, fast, and suitable for offline apps |
| **hive\_flutter** | Hive integration | Adds smooth Flutter support and initialization |
| **flutter/material.dart** | UI framework | For Material Design widgets and icons |

No external APIs were required since the app functions fully offline.

**6. Issues and Bugs Encountered and Resolved**

| **Issue** | **Description** | **Solution** |
| --- | --- | --- |
| **Gradle build failure** | App failed to compile due to mismatched Android NDK versions | Updated ndkVersion in build.gradle.kts |
| **Icon not visible** | Material icons were missing | Added uses-material-design: true in pubspec.yaml |
| **Due date not showing** | UI didn’t refresh after picking date | Fixed by adding setState() in date picker |
| **Description missing in task list** | Description not rendered below title | Updated TaskTile widget to include description |

After resolving these issues, the app now runs smoothly across devices.