

**Department of Software Engineering , MUET Jamshoro**

**Subject: Mobile Application Development**

**Complex Engineering Problem ( CEP ) Report**

**Submitted by: 22SW052, 22SW083**

**Section: 1**

## ****SleepSync – Sleep Quality Tracker****

### ****1. Real World Problem Identification****

Many people struggle to maintain a consistent sleep schedule and don’t realize how much their daily sleep duration varies. While fitness trackers exist, most of them are paid or require hardware devices. For an average smartphone user, there is no simple and free way to record sleep hours, check sleep quality, and analyze their sleep patterns visually.  
This inspired the idea of SleepSync — a lightweight and easy-to-use mobile app that helps users keep track of their sleep without needing extra devices or sign-ups.

### ****2. Proposed Solution****

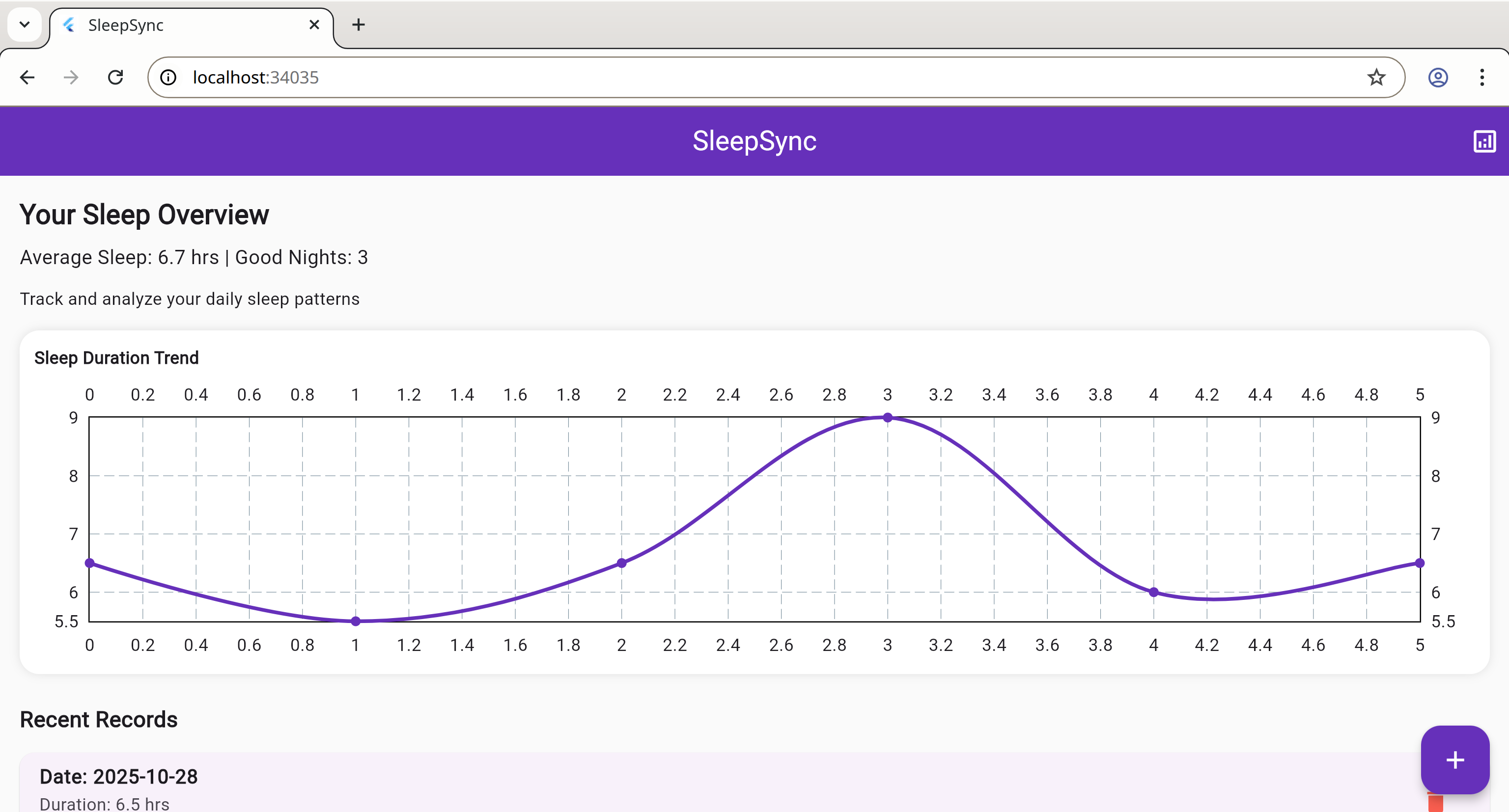
SleepSync solves this problem by allowing users to manually record their sleep duration and quality each day. The app then stores this data locally and presents it in a clean, visual format using charts and summaries.  
Users can easily see how well they are sleeping, identify trends, and even view insights such as average sleep, number of good nights, and best sleep streak.  
The main goal was to build a small but complete Flutter application that meets the basic requirements — multiple screens, database integration, charts, and responsive UI — while being meaningful and practical.

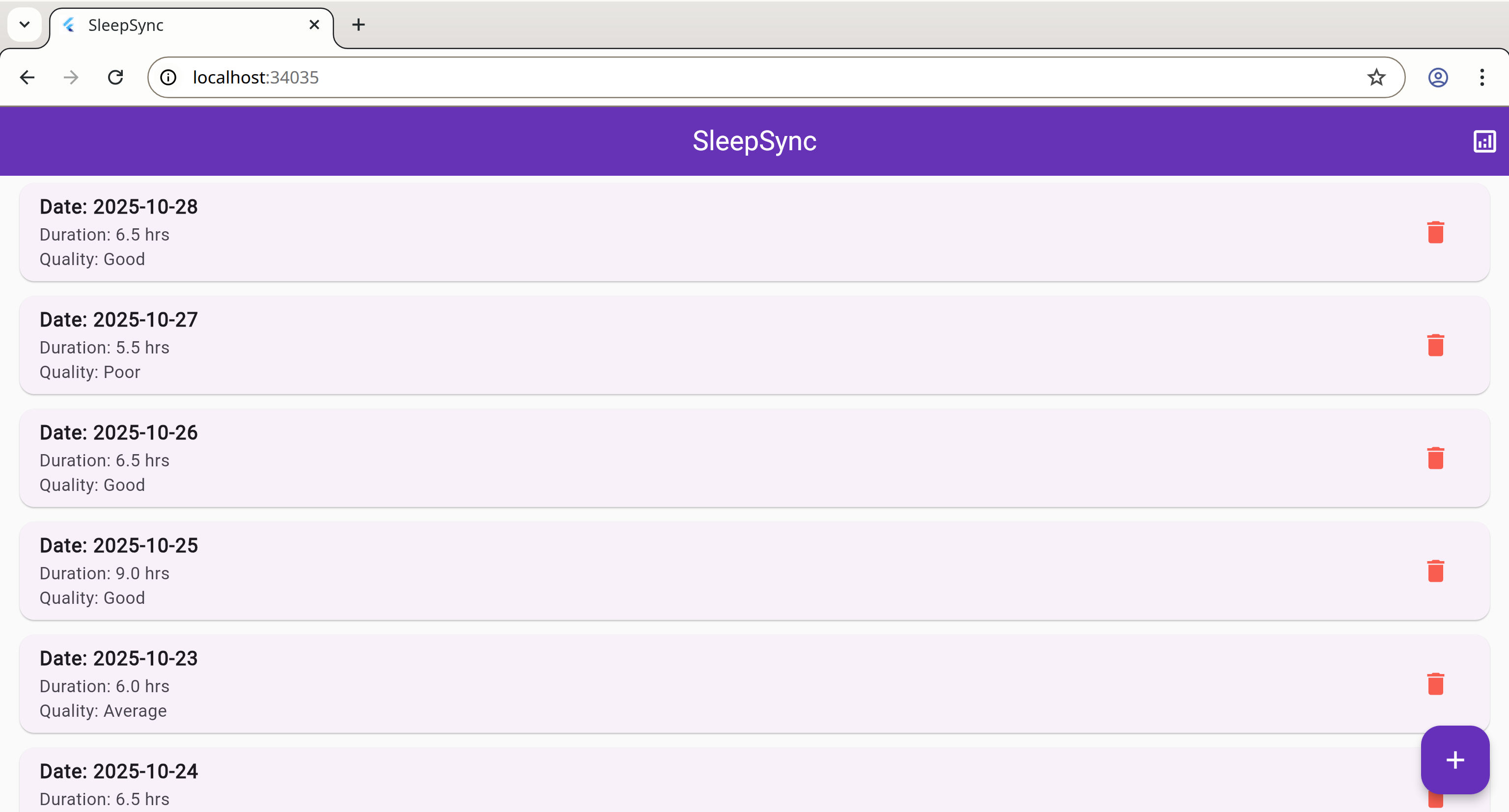
### ****3. Responsive User Interfaces****

### The app has a simple and responsive interface that adapts well to different screen sizes.

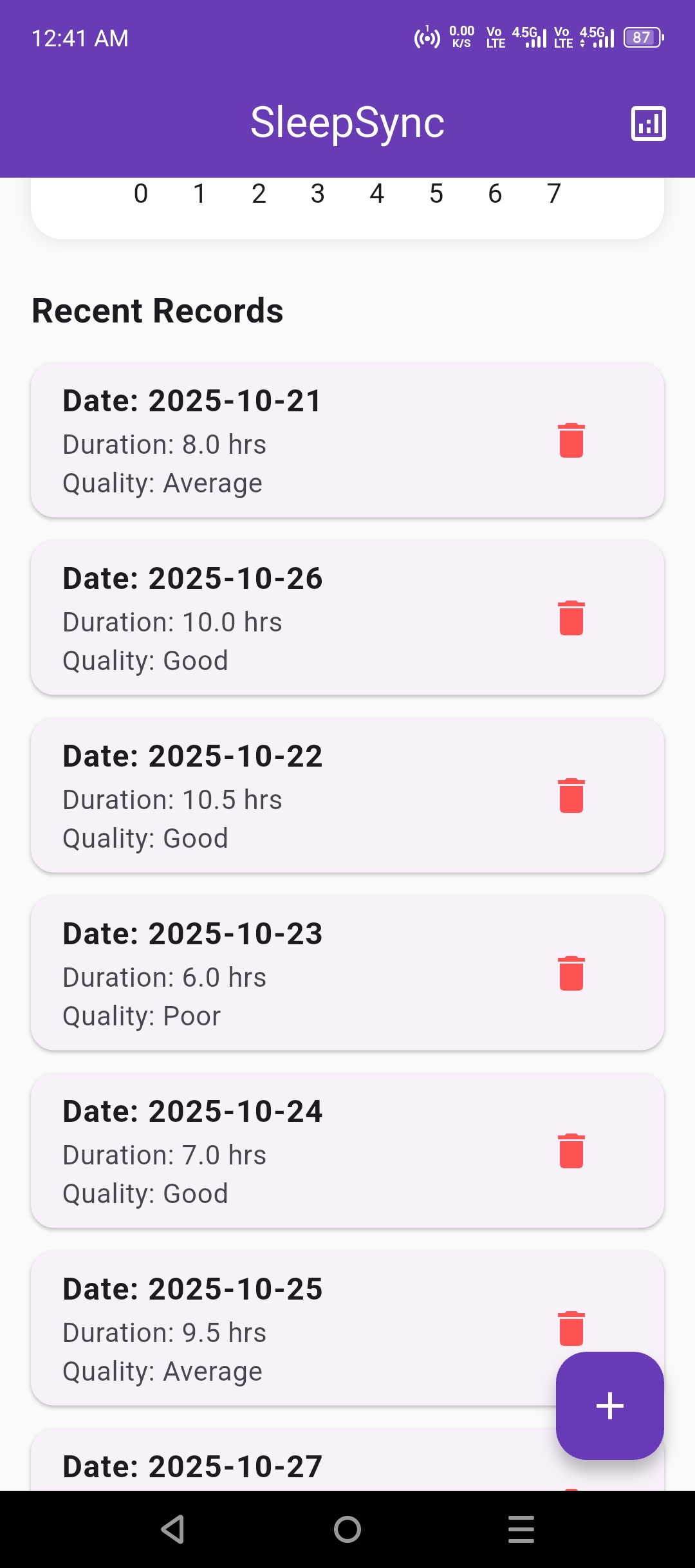
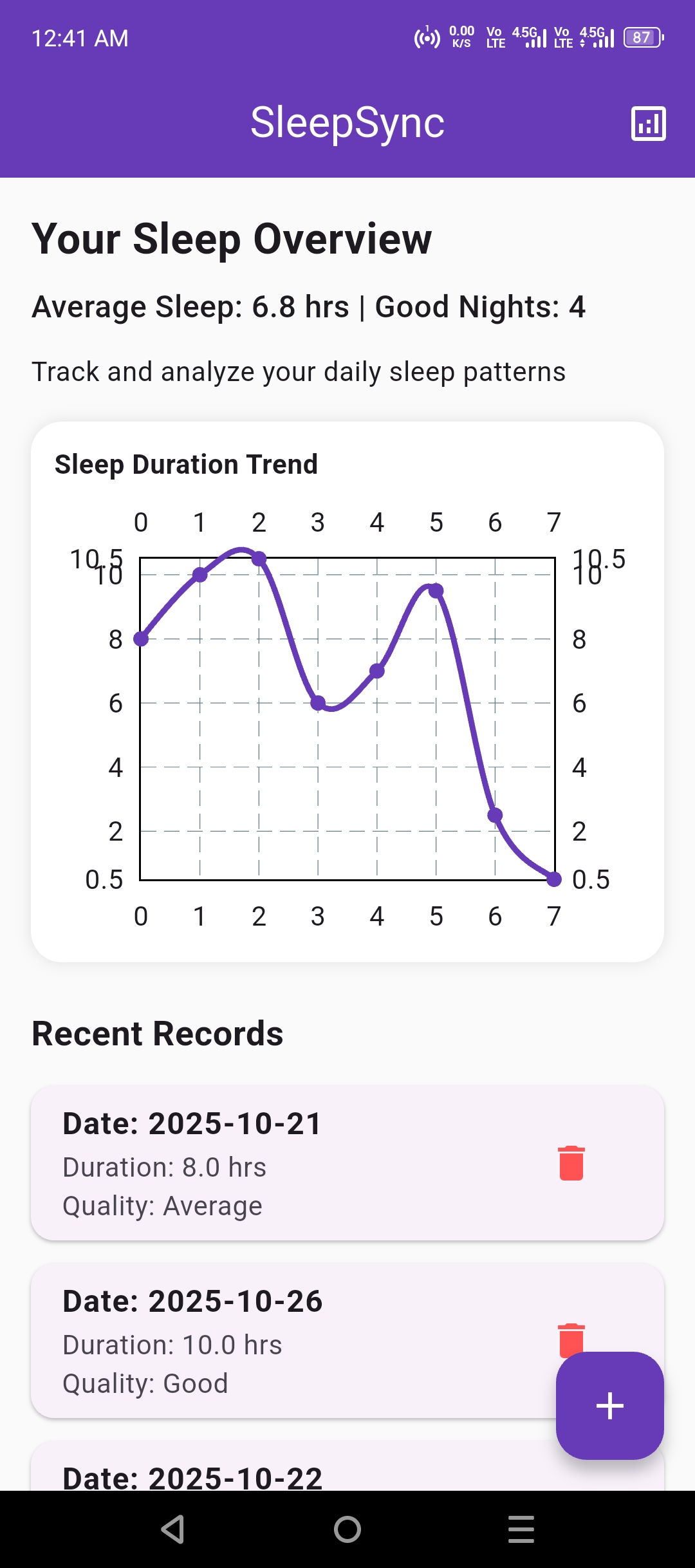
* **Home Screen:** Displays recent records, a summary, and a chart of sleep duration.

**Desktop:**

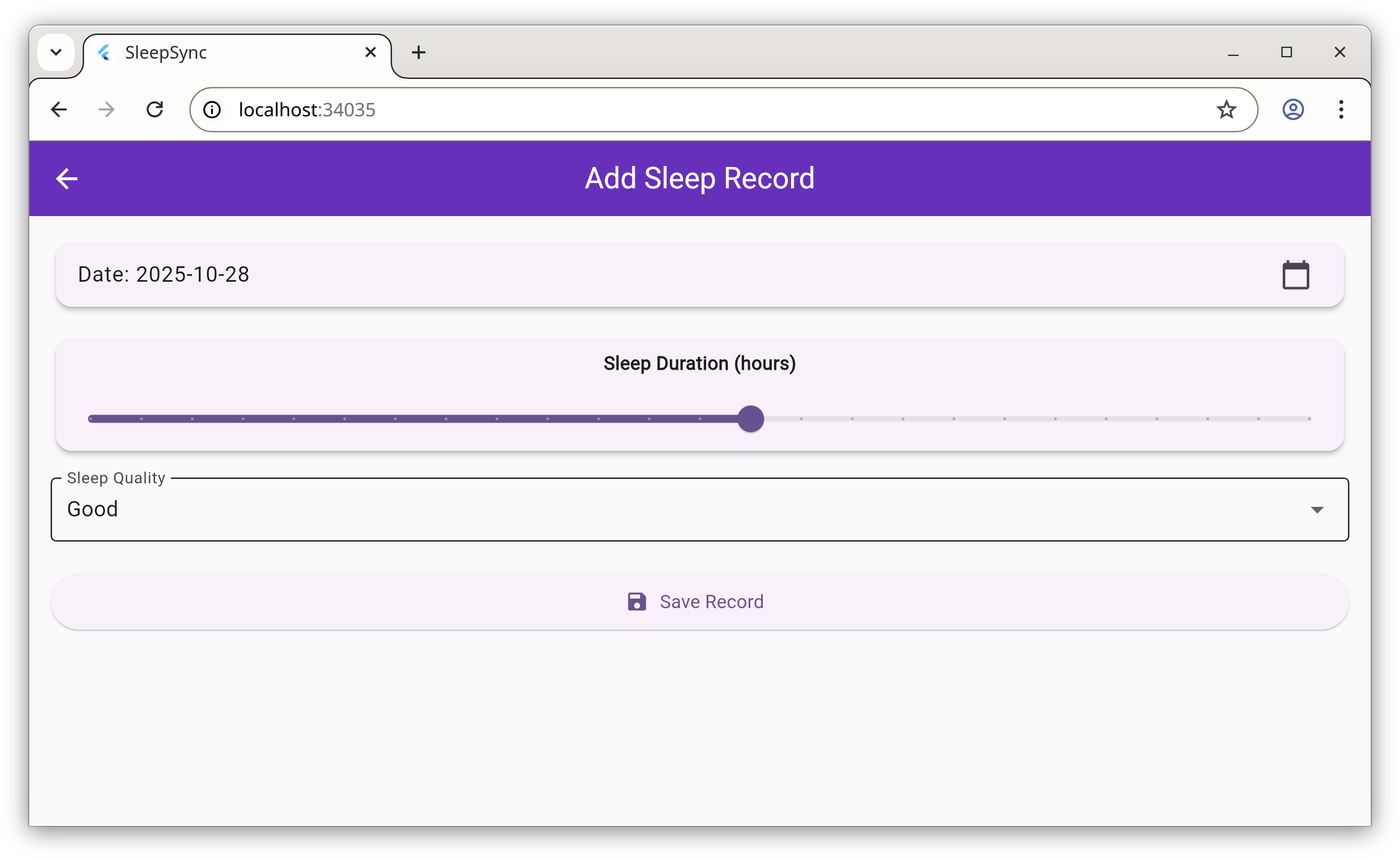




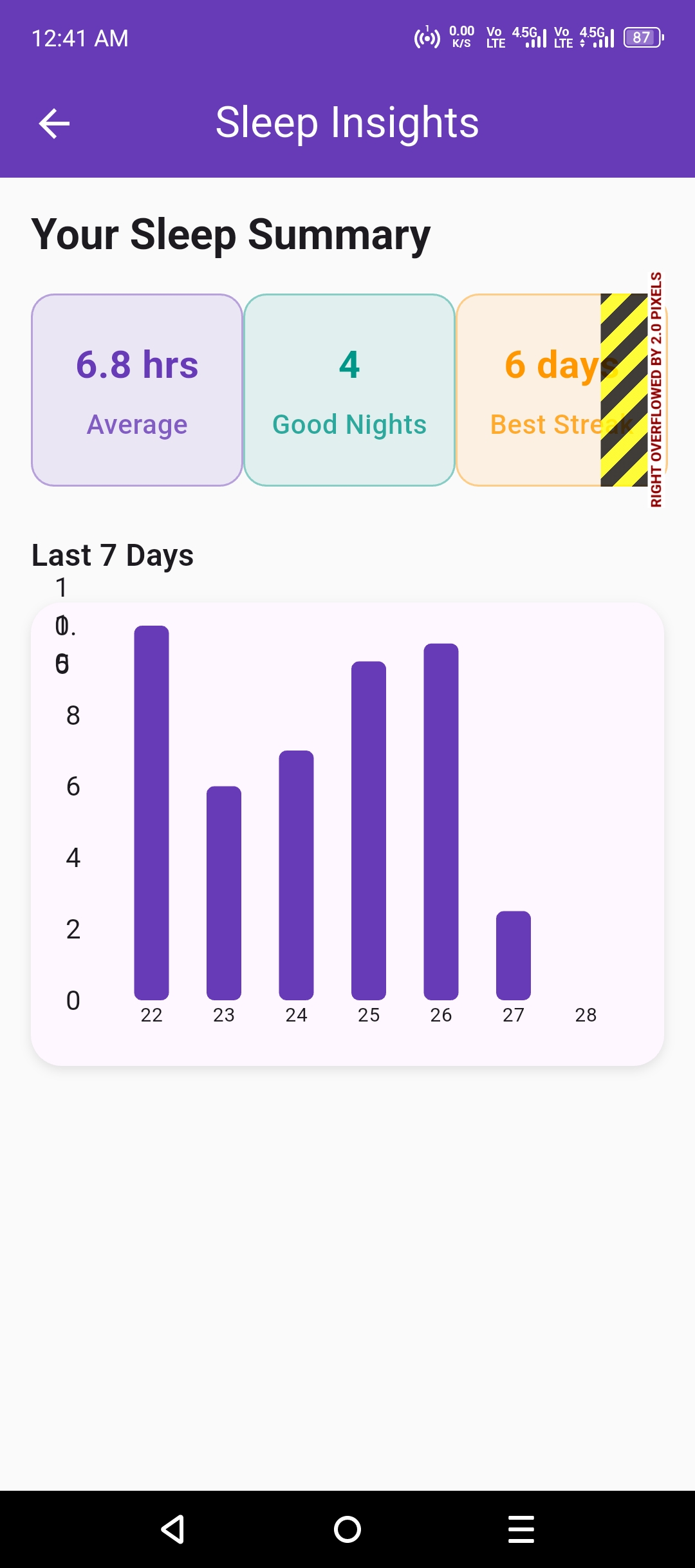
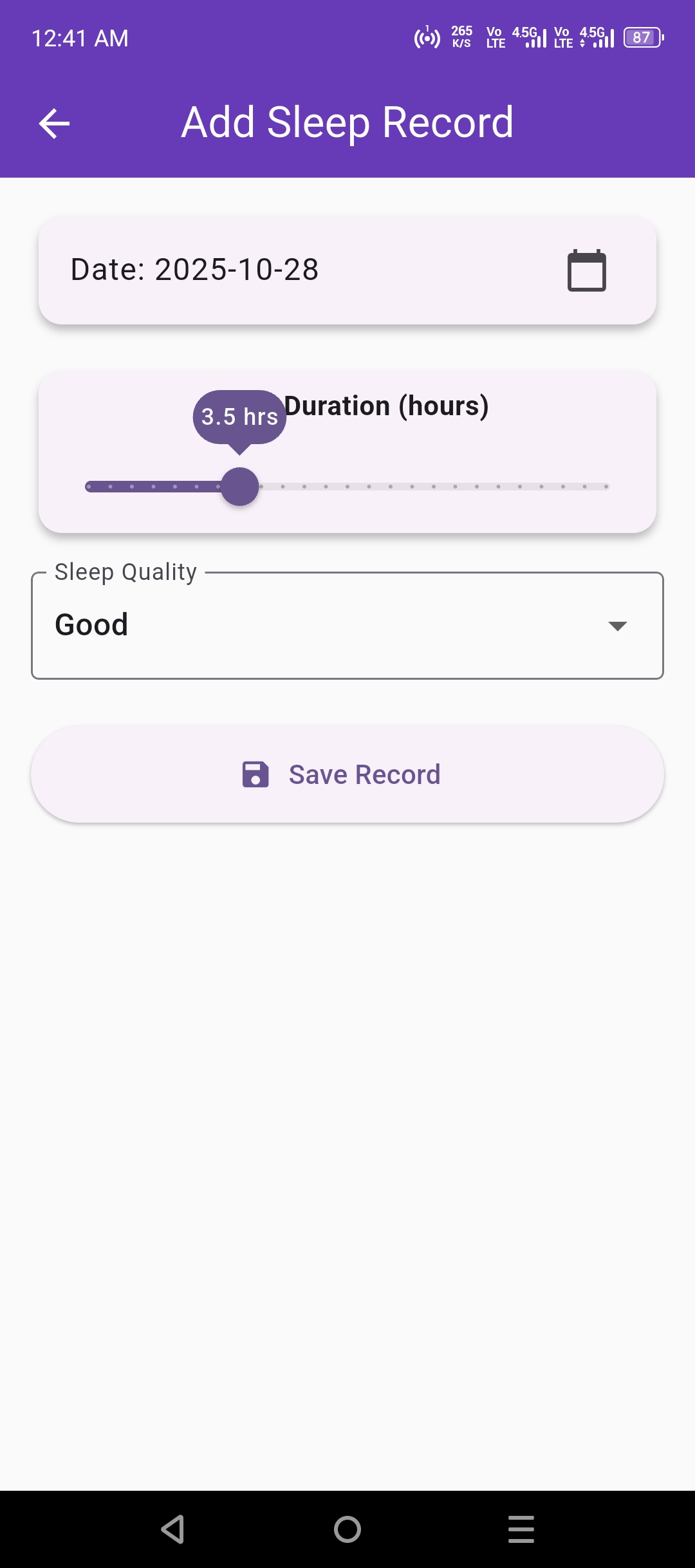
**Android Phone:**



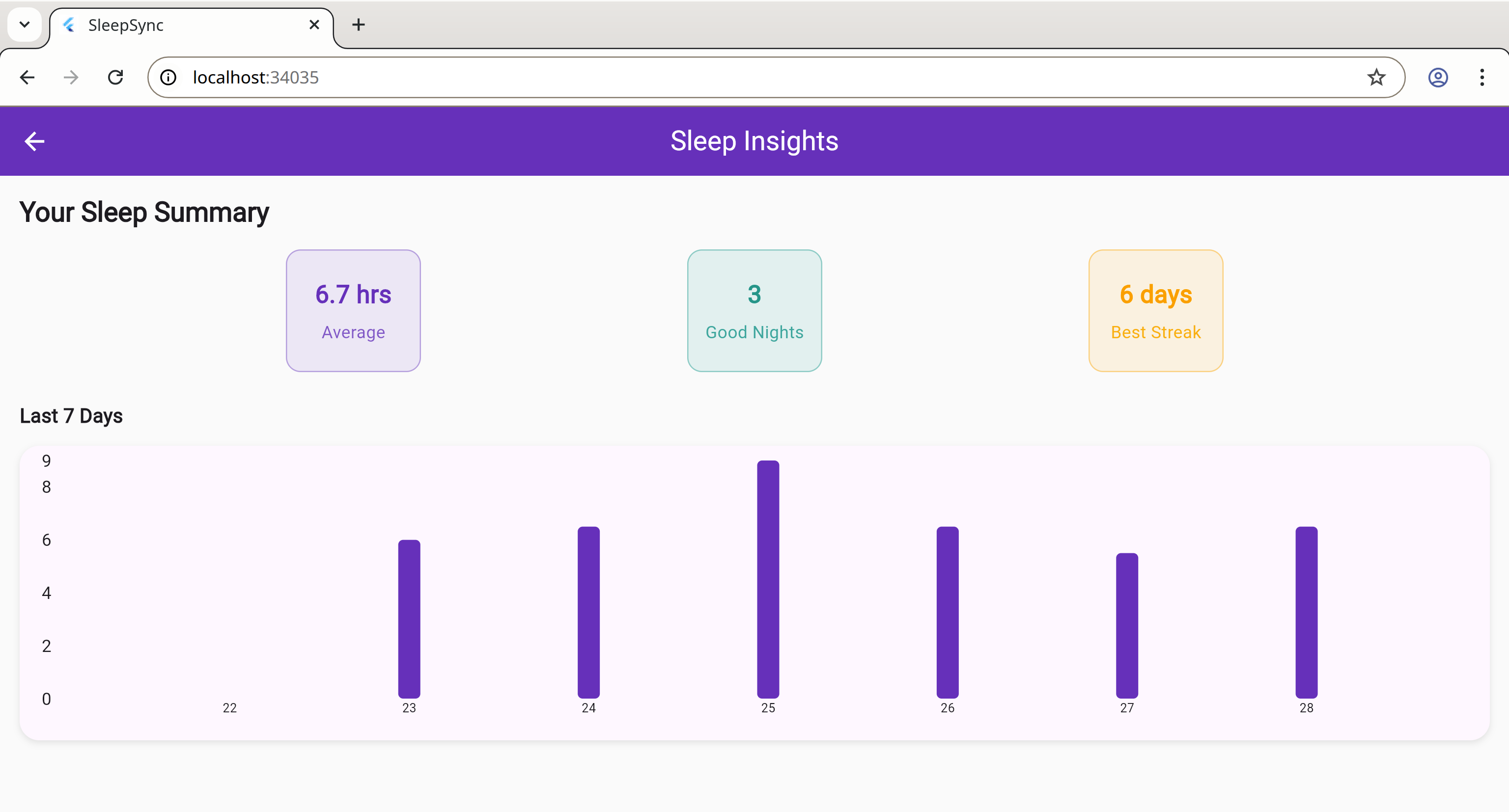
* **Add Record Screen:** Allows the user to enter sleep data (date, duration, and quality).

**Desktop:**

Android Phone:



* **Insights Screen:** Shows statistics and a bar chart for recent sleep activity.  
  The UI is built using Flutter’s Material Design components, with rounded cards, spacing, and adaptive scroll behavior. The design works smoothly on both Android and web platforms.



### ****4. Data Storage****

SleepSync uses the **Hive** database for storing user sleep records locally.  
Hive was chosen because:

* It is **lightweight**, fast, and perfect for small apps.
* It works **offline** and doesn’t require any external setup or internet connection.
* It supports storing **custom objects** (like the SleepRecord model) easily.
* It is widely used in Flutter apps for CEP-level projects due to its simplicity.

The database structure includes one box (sleep\_records) that stores objects containing date, duration, and quality. The data is managed through a SleepProvider class using the Provider package, which keeps the UI and data synchronized.

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

| Package | Purpose | Justification |
| --- | --- | --- |
| **provider** | State management | Keeps the app’s data and UI in sync efficiently. |
| **hive** & **hive\_flutter** | Local database | Provides offline data storage without internet dependency. |
| **fl\_chart** | Data visualization | Used to display sleep duration and trends through bar charts. |

All these packages are open-source, lightweight, and suitable for small educational projects. They simplify app development while keeping the performance fast.

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

| ****Issue**** | ****Description**** | ****Solution**** |
| --- | --- | --- |
| **Hive adapter not found** | The generated file sleep\_record.g.dart was missing, causing build errors during compilation. | Fixed by running flutter pub run build\_runner build --delete-conflicting-outputs to regenerate the adapter. |
| **Chart height too short** | The sleep duration chart appeared squished vertically and didn’t display bars clearly. | Increased container height and adjusted padding for better visualization. |
| **Scroll overflow error** | The Home screen content overflowed when too many records were added. | Wrapped the main column in a SingleChildScrollView to make the full screen scrollable. |
| **State not updating after adding records** | Newly added sleep records didn’t appear immediately on the Home screen. | Used the Provider notifyListeners() method to refresh the UI whenever records changed. |
| **Date picker showing wrong format** | The selected date in Add Record screen appeared in a confusing format. | Used DateFormat('dd-MM-yyyy') from the intl package to display a readable date format. |
| **Database initialization delay** | Hive database took time to initialize, causing a blank screen on startup. | Added await Hive.initFlutter() before running the app to ensure the database loads properly before UI rendering. |
| **Data not persisting across app restarts** | Sleep records were disappearing after restarting the app due to incorrect Hive box reference. | Opened the Hive box using a consistent name (sleep\_records) in both Provider and main file to maintain data persistence. |

**Final Note:**  
SleepSync has responsive multi-screen UI, database integration, and working functionality. It’s simple, practical, and polished enough to demonstrate both technical and creative aspects of mobile app development.