Flutter Habit Tracker App - Task Document

Objective

Build a Flutter-based Habit Tracker within 2 hours to showcase:

- Advanced UI design
- Complex logic handling
- Clean architecture
- Constructive thinking

UI Design Requirements

Main Screen:

- Title: Habit Tracker Screen
- Layout: 2-column responsive grid
- Card Contents per Habit:
- Habit Name (bold text)
- Streak Count (e.g., 3 days)
- Badge for 0-streak habits (icon)
- Floating Action Button (FAB) to add new habits

Styling:

- Gradient background
- Subtle animations
- Material Design compliance
- Accessibility: Min 48x48px targets, color contrast, semantic labels

Functionality

Habit Model:

```
class Habit {
final String name;
int streak;
DateTime? lastCompleted;

Habit({required this.name, this.streak = 0, this.lastCompleted});
}
```

Features:

- Add Habit (dialog with validations)
- Mark as Completed (once/day)
- Reset if not done today
- Delete Habit
- Display Grid: Sorted by streak, 2 columns
- Badge on streak = 0

Logic Considerations

- Validate name (non-empty, max 20 characters)
- Date Check Function:

```
bool isSameDay(DateTime a, DateTime b) => a.year == b.year && a.month == b.month && a.day == b.day;
```

- Sort by streak descending
- Handle empty grid, day change

Technical Stack

- Flutter SDK
- State Management: Provider
- Folder Structure:
- models/, providers/, widgets/, screens/
- Persistence: In-memory list
- Animations: AnimatedScale, AnimatedContainer

Folder Structure

```
lib/
■■■ main.dart
■■■ models/
■ ■■■ habit.dart
■■■ providers/
■ ■■■ habit_provider.dart
■■■ widgets/
■ ■■■ habit_card.dart
■ ■■■ add_habit_dialog.dart
■■■ screens/
■■■ habit_tracker_screen.dart
```

Getting Started

Prerequisites:

- Flutter SDK
- IDE (VS Code, Android Studio)

Setup:

```
$ git clone <repo-url>
$ cd flutter-habit-tracker
$ flutter pub get
$ flutter run
```

Design Decisions

Feature	Decision
State Management	Provider
Persistence	In-memory
UI Grid	GridView.builder
Animations	AnimatedScale
Accessibility	Contrast check, tap size

Evaluation Checklist

Area	Criteria
UI Skills	Grid, animations, contrast
Logic	Streak logic, validations
Architecture	Modular, reusable widgets
Edge Handling	Empty state, invalid input

Time Limit

Total Time: 2 hours
- UI & Layout: 40 min
- State & Logic: 50 min
- Animations & Edge Cases: 20 min
- Testing & Cleanup: 10 min