Assignment 01 — To-Do List App

Mobile Computing Lab

Helwan University — Faculty of Engineering Computer Engineering

Habeba Mostafa Desoky

---

1. Project Overview

This report documents the development of a To-Do List mobile application built using Flutter. The app allows users to manage their daily goals by adding and deleting items from a dynamic list.

2. Color Palette

The app uses a consistent Material Design-inspired blue color palette throughout all components. The following table outlines the colors used:

Color Name	Hex Code	Usage
Primary Blue	`#1565C0`	App bar, buttons, highlights
Light Blue	`#BBDEFB`	Avatar backgrounds, accents
White	`#FFFFFF`	Background, card surfaces
Dark Text	`#212121`	Body text, list items
Grey	`#9E9E9E`	Placeholder text, secondary info
Red	`#F44336`	Delete icon / destructive actions

3. Fonts Used

Primary Font: MyCustomFont

A custom font is applied throughout the app for titles and body text, providing a distinctive and consistent typographic identity. It is loaded via the Flutter font configuration in pubspec.yaml.

System Fallback

Where the custom font is unavailable, the system default sans-serif font is used as a fallback to maintain readability.

4. App Components

The application is structured around the following key UI components:

- Text (Title): Displays 'What do you want to achieve today?' at the top of the screen.
- TextField: Allows the user to type in a new goal. Supports keyboard submission via onSubmitted.
- ElevatedButton (Add Goal): Triggers the _addGoal() function to add the typed goal to the list.
- ListView.builder: Renders the list of goals as Cards, supporting smooth scrolling for any number of items.
- IconButton (Delete): Each list item has a red delete icon that removes it from the list via setState().

5. State Management

The app uses Flutter's built-in StatefulWidget with two state variables:

- _goalController (TextEditingController): Tracks the current text typed in the input field and clears it after a goal is added.
- _goals (List<String>): Stores all added goals. UI rebuilds automatically whenever this list is modified via setState().

6. Core Function: _addGoal()

The _addGoal() function handles the button click event:

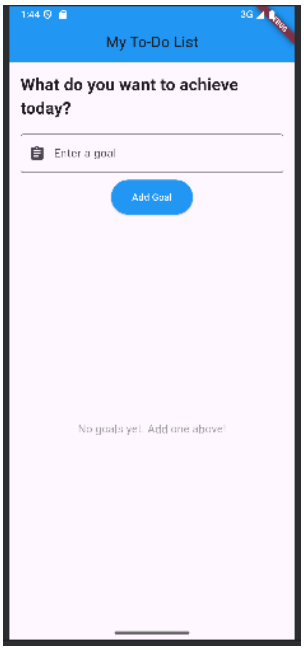
- Reads the current value from _goalController and trims whitespace.
- Validates that the input is not empty.
- Adds the new goal to the _goals list using setState() to trigger a UI rebuild.
- Clears the text field after adding.

7. App Screenshots

The following screenshots illustrate the app's different states and interactions:

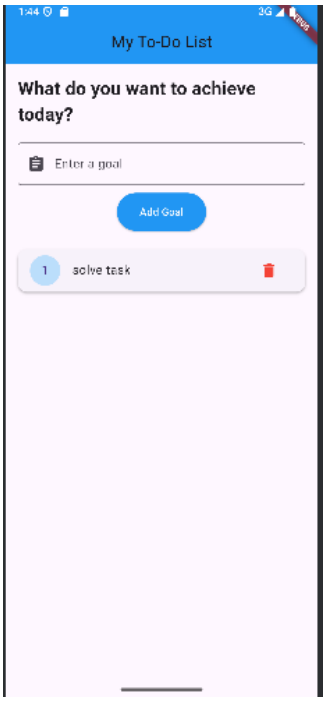
Screenshot 1: Empty State

When no goals have been added, the list area displays: 'No goals yet. Add one above!'



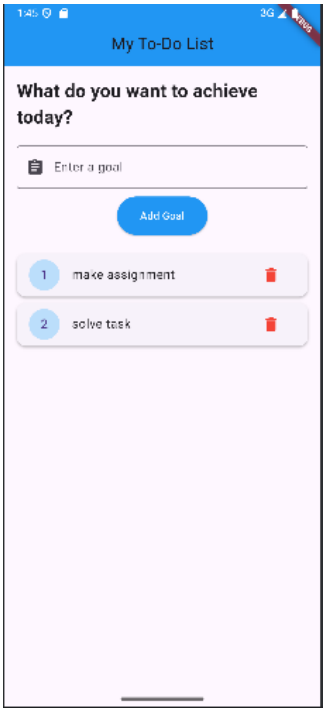
Screenshot 2: Adding a Goal

The user types a goal into the TextField. The 'Add Goal' button becomes the primary call-to-action.



Screenshot 3: List with Multiple Goals

Goals appear as Cards in the ListView. Each card shows a numbered CircleAvatar, the goal text, and a delete icon.



8. Project Links

GitHub Repository

GitHub Repo	https://github.com/habebamostafa/Mobile-Computing-/tree/main/lab1_ToDo/to_do

9. Demo Video

A 2-minute demo video was recorded showing the app in runtime

https://drive.google.com/file/d/1XRrwNuFoth-QHI1hDJn52X2j5Wm3Tn8-/view?usp=sharing