import 'package:flutter/material.dart';

void main() {

runApp(const ToDoApp());

}

class ToDoApp extends StatelessWidget {

const ToDoApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'To-Do List App',

debugShowCheckedModeBanner: false,

theme: ThemeData(

primarySwatch: Colors.deepPurple,

useMaterial3: true,

),

home: const ToDoHomePage(),

);

}

}

class ToDoHomePage extends StatefulWidget {

const ToDoHomePage({super.key});

@override

State<ToDoHomePage> createState() => \_ToDoHomePageState();

}

class \_ToDoHomePageState extends State<ToDoHomePage> {

final TextEditingController \_taskController = TextEditingController();

final List<Map<String, dynamic>> \_tasks = [];

void \_addTask() {

String task = \_taskController.text.trim();

if (task.isNotEmpty) {

setState(() {

\_tasks.add({'title': task, 'done': false});

});

\_taskController.clear();

}

}

void \_toggleTask(int index) {

setState(() {

\_tasks[index]['done'] = !\_tasks[index]['done'];

});

}

void \_deleteTask(int index) {

setState(() {

\_tasks.removeAt(index);

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('📝 To-Do List'),

centerTitle: true,

backgroundColor: Colors.deepPurple,

foregroundColor: Colors.white,

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

children: [

// Input field and button

Row(

children: [

Expanded(

child: TextField(

controller: \_taskController,

decoration: const InputDecoration(

hintText: 'Enter a new task...',

border: OutlineInputBorder(),

),

),

),

const SizedBox(width: 8),

ElevatedButton(

onPressed: \_addTask,

child: const Text('Add'),

),

],

),

const SizedBox(height: 16),

// Task list

Expanded(

child: \_tasks.isEmpty

? const Center(

child: Text(

'No tasks yet! Add something to get started 😊',

style: TextStyle(fontSize: 16),

textAlign: TextAlign.center,

),

)

: ListView.builder(

itemCount: \_tasks.length,

itemBuilder: (context, index) {

final task = \_tasks[index];

return Card(

margin: const EdgeInsets.symmetric(vertical: 6),

child: ListTile(

leading: Checkbox(

value: task['done'],

onChanged: (\_) => \_toggleTask(index),

),

title: Text(

task['title'],

style: TextStyle(

fontSize: 16,

decoration: task['done']

? TextDecoration.lineThrough

: null,

color: task['done']

? Colors.grey

: Colors.black,

),

),

trailing: IconButton(

icon: const Icon(Icons.delete, color: Colors.red),

onPressed: () => \_deleteTask(index),

),

),

);

},

),

),

],

),

),

);

}

}