

```

import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: ToDoListScreen(),
    );
  }
}

class ToDoListScreen extends StatefulWidget {
  const ToDoListScreen({super.key});

  @override
  ToDoListScreenState createState() => ToDoListScreenState();
}

class ToDoListScreenState extends State<ToDoListScreen> {
  List<Map<String, dynamic>> tasks = [];
  TextEditingController controller = TextEditingController();

  void addTask(String task) {
    if (task.isNotEmpty) {
      tasks.add({'task': task, 'completed': false});
      controller.clear();
      setState(() {});
    }
  }

  void toggleTaskCompletion(int index) {
    tasks[index]['completed'] = !tasks[index]['completed'];
    setState(() {});
  }

  void deleteTask(int index) {
    tasks.removeAt(index);
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('To-Do List')),
      body: Column(
        children: [
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Row(
              children: [
                Expanded(
                  child: TextField(
                    controller: controller,
                    decoration: InputDecoration(labelText: 'Enter Task'),
                  ),
                IconButton(
                  icon: Icon(Icons.add),
                  onPressed: () => addTask(controller.text),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}

```

```

        ),
    ],
),
Expanded(
    child: tasks.isEmpty
        ? Center(child: Text('No tasks yet!'))
        : ListView.builder(
            itemCount: tasks.length,
            itemBuilder: (context, index) {
                var task = tasks[index];
                return ListTile(
                    title: Text(
                        task['task'],
                        style: TextStyle(
                            decoration: task['completed']
                                ? TextDecoration.lineThrough
                                : TextDecoration.none,
                        ),
                    ),
                    leading: Checkbox(
                        value: task['completed'],
                        onChanged: (_) => toggleTaskCompletion(index),
                    ),
                    trailing: IconButton(
                        icon: Icon(Icons.delete, color: Colors.red),
                        onPressed: () => deleteTask(index),
                    ),
                ),
            ),
        );
    },
),
],
),
);
}
}

```