Main.dart

```
import 'package:flutter/material.dart';
import 'package:task8/todo.dart';

import 'package:task8/todoList.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key? key}) : super(key: key);

  late List<todoModel> lst;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    widget.lst = lstTodo;
    return Scaffold(
      appBar: AppBar(
        title: Text("Todo List"),
        actions: const [
          Padding(
            padding: EdgeInsets.only(right: 15),
            child: Icon(Icons.delete_rounded),
          ),
        ],
      ),
      body: todoList_(
        list: widget.lst,
      ),

      floatingActionButton: FloatingActionButton(
        onPressed: () => {},
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ), // This trailing comma makes auto-formatting nicer for build methods.
    );
  }
}
```

todo.dart:

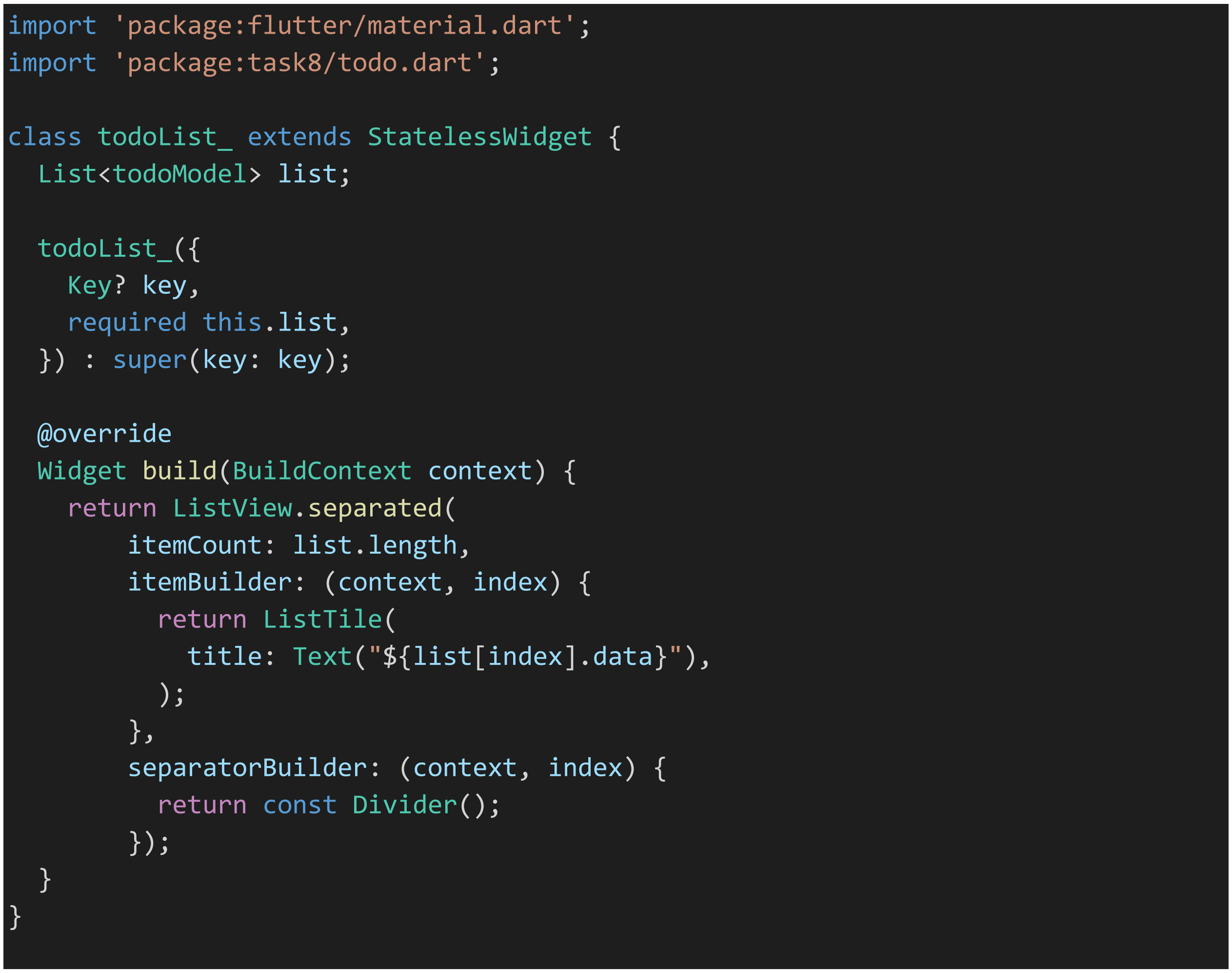
```
class todoModel {
  String data;
  bool status;

  todoModel({required this.data, required this.status});
}

List<todoModel> lstTodo = [
  todoModel(data: "To run program", status: false),
  todoModel(data: "To run program", status: false),
  todoModel(data: "To run program", status: false),
  todoModel(data: "To run program", status: false),
  todoModel(data: "To run program", status: false),
  todoModel(data: "To run program", status: false),
];
```

todoList.dart:

```
// ignore_for_file: file_names
```



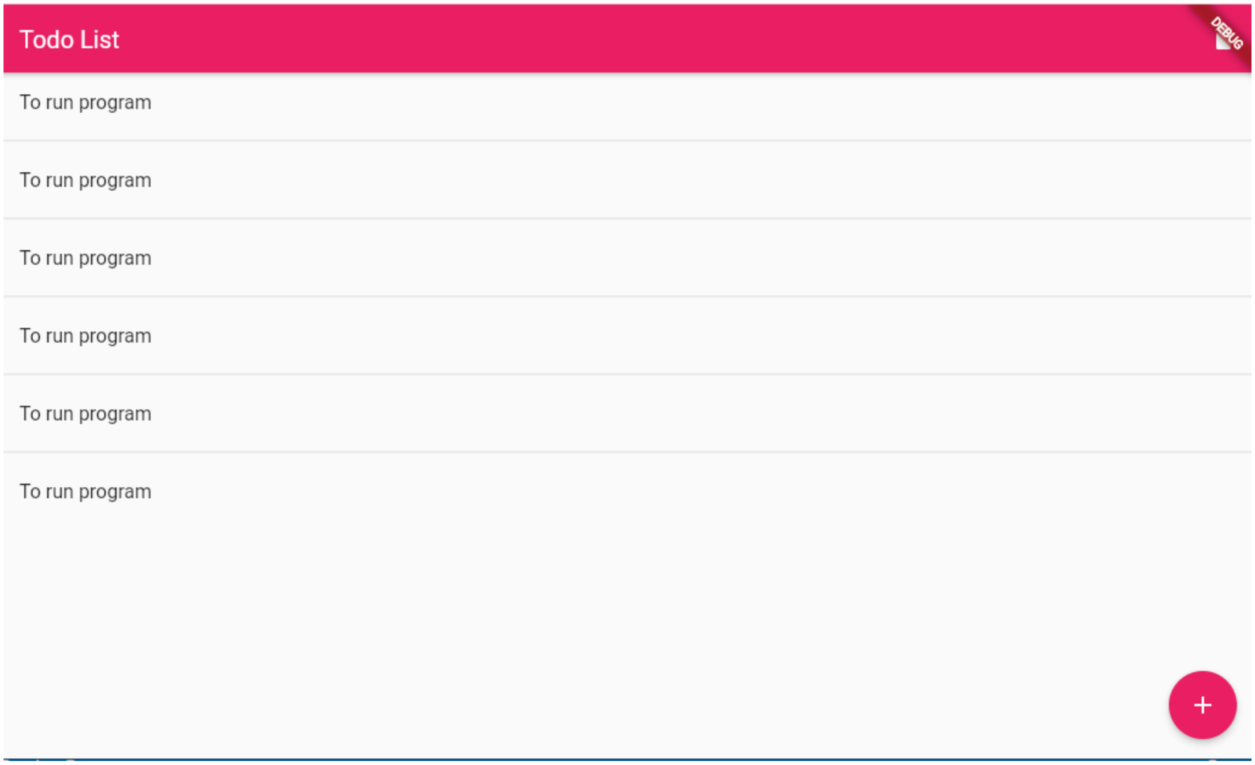
```
import 'package:flutter/material.dart';
import 'package:task8/todo.dart';

class todoList_ extends StatelessWidget {
  List<todoModel> list;

  todoList_({
    Key? key,
    required this.list,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return ListView.separated(
        itemCount: list.length,
        itemBuilder: (context, index) {
          return ListTile(
            title: Text("${list[index].data}"),
          );
        },
        separatorBuilder: (context, index) {
          return const Divider();
        });
  }
}
```

SCREENSHOT:



Todo List

To run program

To run program

To run program

To run program

To run program

To run program