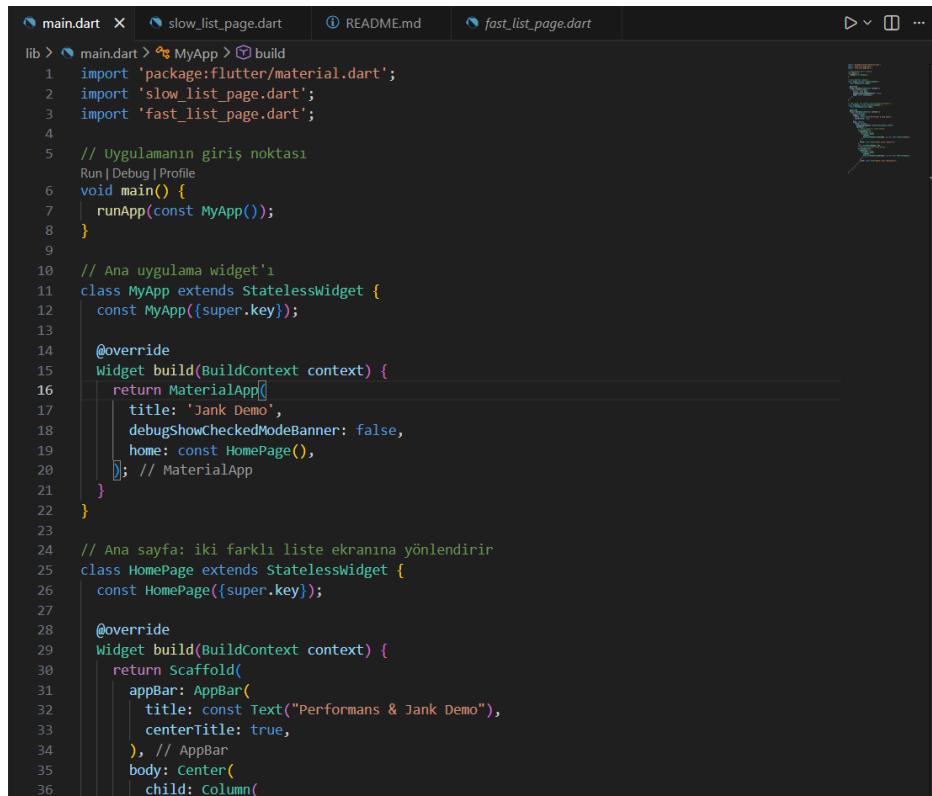
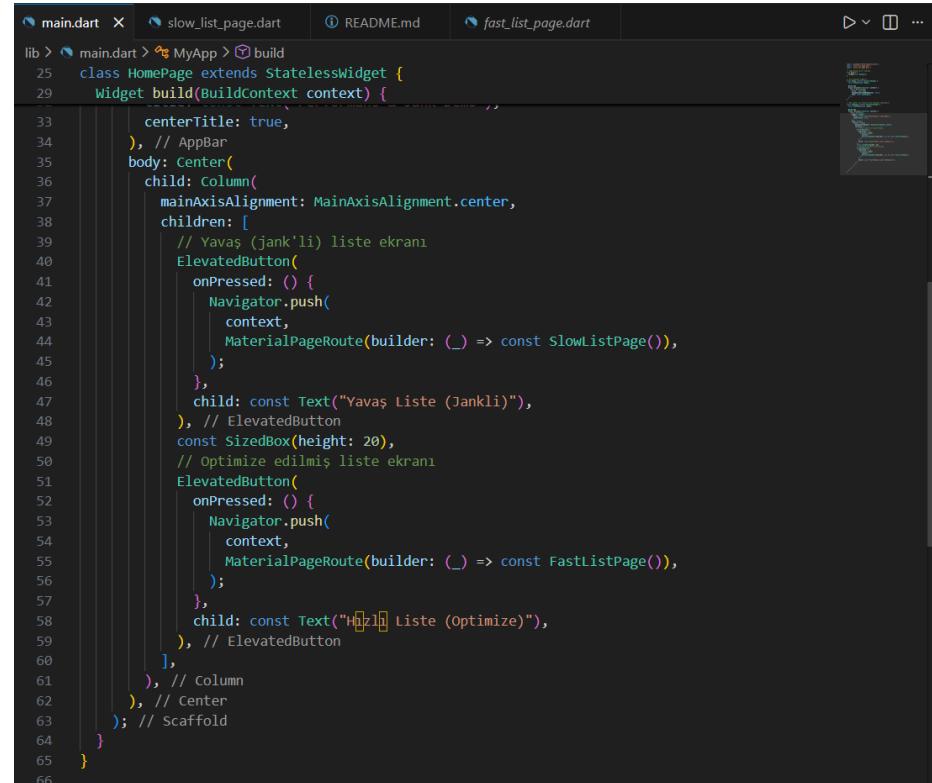


FLUTTER PERFORMANS VE JANK AZALTMA DEMO PROJESİ

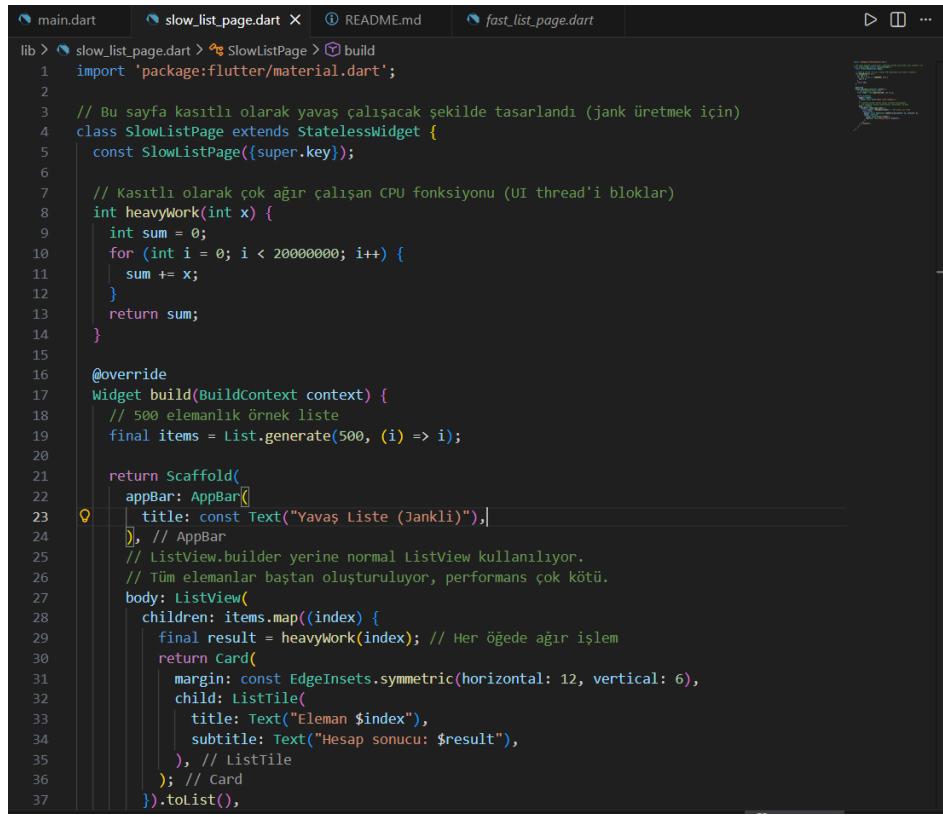
KODLAR:



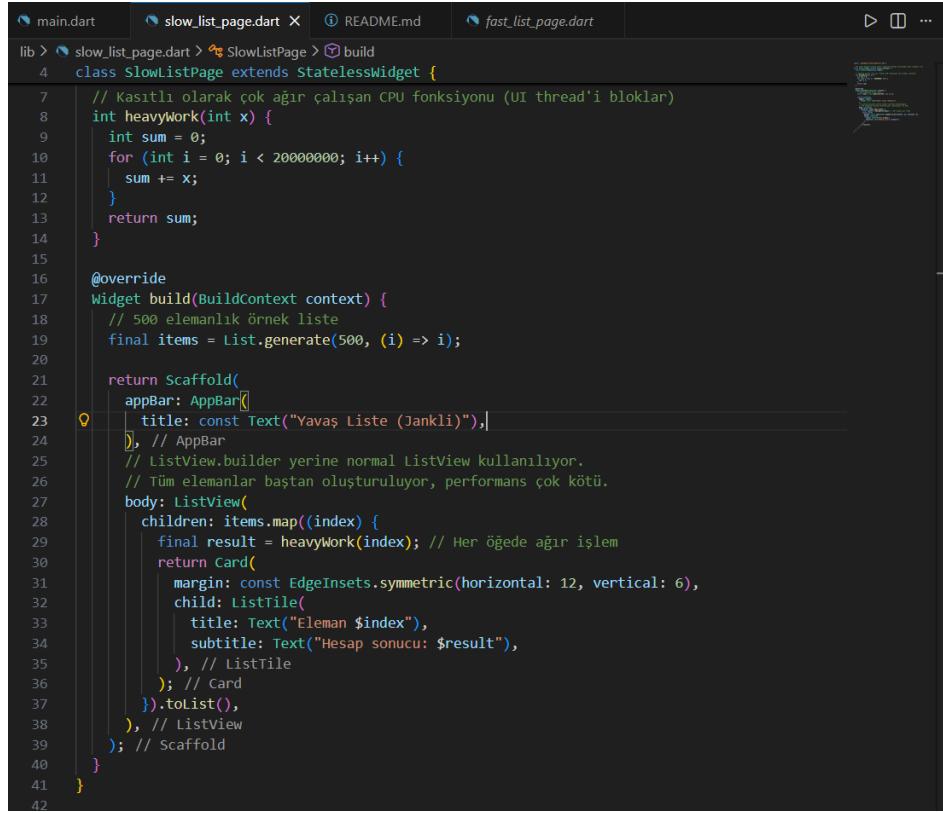
```
lib > lib > MyApp > build
1 import 'package:flutter/material.dart';
2 import 'slow_list_page.dart';
3 import 'fast_list_page.dart';
4
5 // Uygulamanın giriş noktası
6 Run | Debug | Profile
7 void main() {
8   runApp(const MyApp());
9 }
10
11 // Ana uygulama widget'i
12 class MyApp extends StatelessWidget {
13   const MyApp({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     return MaterialApp(
18       title: 'Jank Demo',
19       debugShowCheckedModeBanner: false,
20       home: const HomePage(),
21     ); // MaterialApp
22   }
23
24 // Ana sayfa: iki farklı liste ekranına yönlendirir
25 class HomePage extends StatelessWidget {
26   const HomePage({super.key});
27
28   @override
29   Widget build(BuildContext context) {
30     return Scaffold(
31       appBar: AppBar(
32         title: const Text("Performans & Jank Demo"),
33         centerTitle: true,
34       ), // AppBar
35       body: Center(
36         child: Column(
37           mainAxisAlignment: MainAxisAlignment.center,
38           children: [
39             // Yavaş (jank'lı) liste ekranı
40             ElevatedButton(
41               onPressed: () {
42                 Navigator.push(
43                   context,
44                   MaterialPageRoute(builder: (_) => const SlowListPage()),
45               );
46             },
47             child: const Text("Yavaş Liste (Janklı)"),
48           ), // ElevatedButton
49           const SizedBox(height: 20),
50           // Optimize edilmiş liste ekranı
51           ElevatedButton(
52             onPressed: () {
53               Navigator.push(
54                 context,
55                 MaterialPageRoute(builder: (_) => const FastListPage()),
56               );
57             },
58             child: const Text("Hızlı Liste (Optimize)"),
59           ), // ElevatedButton
60         ],
61       ), // Column
62     ), // Center
63   ); // Scaffold
64 }
65
66 }
```



```
lib > lib > MyApp > build
25 class HomePage extends StatelessWidget {
29   Widget build(BuildContext context) {
30     return Scaffold(
31       appBar: AppBar(
32         title: const Text("Performans & Jank Demo"),
33         centerTitle: true,
34       ), // AppBar
35       body: Center(
36         child: Column(
37           mainAxisAlignment: MainAxisAlignment.center,
38           children: [
39             // Yavaş (jank'lı) liste ekranı
40             ElevatedButton(
41               onPressed: () {
42                 Navigator.push(
43                   context,
44                   MaterialPageRoute(builder: (_) => const SlowListPage()),
45               );
46             },
47             child: const Text("Yavaş Liste (Janklı)"),
48           ), // ElevatedButton
49           const SizedBox(height: 20),
50           // Optimize edilmiş liste ekranı
51           ElevatedButton(
52             onPressed: () {
53               Navigator.push(
54                 context,
55                 MaterialPageRoute(builder: (_) => const FastListPage()),
56               );
57             },
58             child: const Text("Hızlı Liste (Optimize)"),
59           ), // ElevatedButton
60         ],
61       ), // Column
62     ); // Scaffold
63   }
64 }
```



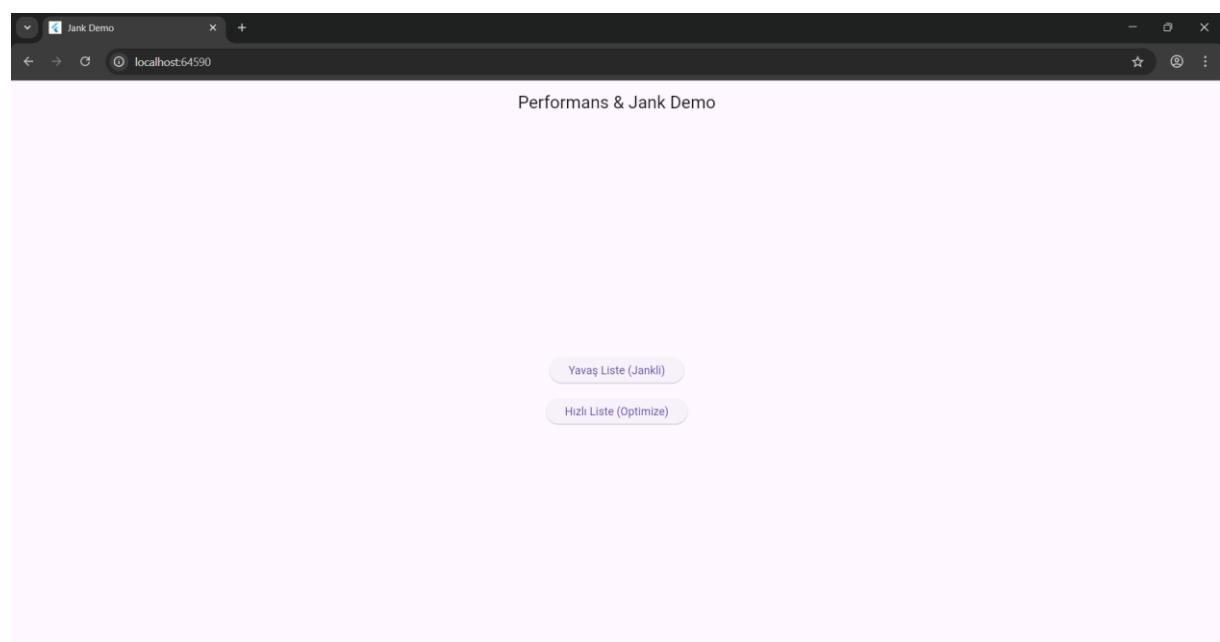
```
lib > slow_list_page.dart > SlowListPage > build
1 import 'package:flutter/material.dart';
2
3 // Bu sayfa kasitli olarak yavaş çalışacak şekilde tasarlandı (jank üretmek için)
4 class SlowListPage extends StatelessWidget {
5   const SlowListPage({super.key});
6
7   // Kasitli olarak çok ağır çalışan CPU fonksiyonu (UI thread'i bloklar)
8   int heavyWork(int x) {
9     int sum = 0;
10    for (int i = 0; i < 20000000; i++) {
11      sum += x;
12    }
13    return sum;
14  }
15
16  @override
17  Widget build(BuildContext context) {
18    // 500 elemanlık örnek liste
19    final items = List.generate(500, (i) => i);
20
21    return Scaffold(
22      appBar: AppBar(
23        title: const Text("Yavaş Liste (Janklı)"),
24      ), // AppBar
25      // ListView.builder yerine normal ListView kullanılıyor.
26      // Tüm elemanlar baştan oluşturuluyor, performans çok kötü.
27      body: ListView(
28        children: items.map((index) {
29          final result = heavyWork(index); // Her öğede ağır işlem
30          return Card(
31            margin: const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
32            child: ListTile(
33              title: Text("Eleman $index"),
34              subtitle: Text("Hesap sonucu: $result"),
35            ), // ListTile
36          ); // Card
37        }).toList(),
38      ), // ListView
39    ); // Scaffold
40  }
41}
```



```
lib > slow_list_page.dart > SlowListPage > build
4 class SlowListPage extends StatelessWidget {
7   // Kasitli olarak çok ağır çalışan CPU fonksiyonu (UI thread'i bloklar)
8   int heavyWork(int x) {
9     int sum = 0;
10    for (int i = 0; i < 20000000; i++) {
11      sum += x;
12    }
13    return sum;
14  }
15
16  @override
17  Widget build(BuildContext context) {
18    // 500 elemanlık örnek liste
19    final items = List.generate(500, (i) => i);
20
21    return Scaffold(
22      appBar: AppBar(
23        title: const Text("Yavaş Liste (Janklı)"),
24      ), // AppBar
25      // ListView.builder yerine normal ListView kullanılıyor.
26      // Tüm elemanlar baştan oluşturuluyor, performans çok kötü.
27      body: ListView(
28        children: items.map((index) {
29          final result = heavyWork(index); // Her öğede ağır işlem
30          return Card(
31            margin: const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
32            child: ListTile(
33              title: Text("Eleman $index"),
34              subtitle: Text("Hesap sonucu: $result"),
35            ), // ListTile
36          ); // Card
37        }).toList(),
38      ), // ListView
39    ); // Scaffold
40  }
41}
```

```
main.dart slow_list_page.dart README.md fast_list_page.dart
lib > fast_list_page.dart > FastListPage > build
1 import 'package:flutter/material.dart';
2
3 // Bu sayfa, optimize edilmiş hızlı liste sürümür.
4 class FastListPage extends StatelessWidget {
5   const FastListPage({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     final items = List.generate(500, (i) => i);
10
11   return Scaffold(
12     appBar: AppBar(
13       title: const Text("Hızlı Liste (Optimize)"),
14     ), // AppBar
15     // ListView.builder, yalnızca ekranda görünen elemanları oluşturur.
16     body: ListView.builder(
17       itemCount: items.length,
18       itemBuilder: (context, index) {
19         return RepaintBoundary(
20           // RepaintBoundary, her satırın ayrı ayrı yeniden çizilmesini sağlar.
21           child: Card(
22             margin: const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
23             child: ListTile(
24               title: Text("Eleman ${index}"),
25               subtitle: const Text("Optimize edilmiş, jank yok"),
26               trailing: const Icon(Icons.check_circle, color: Colors.green),
27             ), // ListTile
28             ), // Card
29           ); // RepaintBoundary
30         },
31       ), // ListView.builder
32     ), // scaffold
33   );
34 }
35
```

EKRAN ÇIKTILARI:



Jank Demo

localhost:64590

← Yavaş Liste (Janklı)

Eleman 0	Hesap sonucu: 0
Eleman 1	Hesap sonucu: 20000000
Eleman 2	Hesap sonucu: 40000000
Eleman 3	Hesap sonucu: 60000000
Eleman 4	Hesap sonucu: 80000000
Eleman 5	Hesap sonucu: 100000000
Eleman 6	Hesap sonucu: 120000000
Eleman 7	Hesap sonucu: 140000000
Eleman 8	Hesap sonucu: 160000000

Jank Demo

localhost:64590

← Hızlı Liste (Optimize)

Eleman 0	Optimize edilmiş, jank yok	✓
Eleman 1	Optimize edilmiş, jank yok	✓
Eleman 2	Optimize edilmiş, jank yok	✓
Eleman 3	Optimize edilmiş, jank yok	✓
Eleman 4	Optimize edilmiş, jank yok	✓
Eleman 5	Optimize edilmiş, jank yok	✓
Eleman 6	Optimize edilmiş, jank yok	✓
Eleman 7	Optimize edilmiş, jank yok	✓
Eleman 8	Optimize edilmiş, jank yok	✓