

Ödev 4-2

⊚ Link	https://youtu.be/BZfClqxRbAl
© Created	@November 26, 2021 1:20 PM

Uygulamanın youtube linki : https://youtu.be/BZfClqxRbAl

Kodlar:

```
import 'package:flutter/material.dart';
import 'package:english_words/english_words.dart';
void main(){
 runApp(MaterialApp(home: MyApp()));
class MyApp extends StatelessWidget {
 Widget build(BuildContext context) {
    return MaterialApp(
     title: 'Startup'
     home: RandomWords(),
   );
 }
}
class RandomWords extends StatefulWidget{
 @override
  _RandomWordsState createState() => _RandomWordsState();
class _RandomWordsState extends State<RandomWords> {
 final _suggestions = <WordPair>[];
final _biggerFont = TextStyle(fontSize: 18);
 @override
 Widget build(BuildContext context) {
    return Scaffold (
      appBar: AppBar(
       title: Text('Startup Adı Oluşturucu'),
      body: _buildSuggestions(),
 Widget _buildSuggestions() {
    return ListView.builder(
                              // her satır için çalış ...
        padding: const EdgeInsets.all(16),
        itemBuilder: (BuildContext _context, int i) {
                                                        //i satır index'ini tutar.
          index 0'dan başlar ve her çalıştırılmada artar.
          Her sözcük eşleştirmesi için iki kez artar: bir kez ListTile için ve bir kez Divider() için.
          Böylece, kullanıcı kaydırdıkça önerilen liste büyümeye devam eder
          if (i.isOdd) { // eğer index tekse ( \ddot{o}rn: 0-> ListTile , 1-> Divider , 2-> ListTile , 3-> Divider . . . )
           return Divider();
                                  // Divider (çizgi) koy
          final int index = i \sim / 2;
          if (index \geq _suggestions.length) { // index , listeden uzunsa ( kullanıcı ekran kaydırıyorsa )
           _suggestions.addAll(generateWordPairs().take(10)); // yenilerini ekle
```

Ödev 4-2

```
return _buildRow(_suggestions[index]);
}
);
}
Widget _buildRow(WordPair pair) { // Satırlar oluşturulur.
   return ListTile(
    title: Text(
      pair.asPascalCase,
      style: _biggerFont,
    ),
   );
}
```

Ekran görüntüleri ve açıklamalar

- · Öncelikle english_words paketini import ettik.
- · MyApp sınıfını çalıştırdık.
- Burada MaterialApp widget'ını kullandık ve RandomWords sınıfımızı çağırdık.

```
import 'package:flutter/material.dart';
         import 'package:english_words/english_words.dart';
  3
       void main(){
  5
          runApp(MaterialApp(home: MyApp()));
 7
       class MyApp extends StatelessWidget {
 8
 9
          @override
 10 01
          Widget build(BuildContext context) {
 11
             return MaterialApp(
               title: 'Startup',
 12
             - home: RandomWords(),
 13
            ); // MaterialApp
 14
 15
          }
       _____}}
 17
 18
       class RandomWords extends StatefulWidget{
 19
 20 0
           _RandomWordsState createState() => _RandomWordsState();
        }
23
       class _RandomWordsState extends State<RandomWords> {
           final _suggestions = <WordPair>[];
24
           final _biggerFont = TextStyle(fontSize: 18);
 25
```

- Bu sınıf StatefulWidget'tan miras almaktadır. State durumu ise _RandomWordsState sınıfında tutulur.
- Bu sınıfta bir liste tutan _suggestions ile fontSize bilgisini tutan _biggerFont değişkenleri var. Liste kelime çiftlerini tutar ve import edilen paket içerisinden alınır.

Ödev 4-2 2

- build metodunda Scaffold widget'ını çağırdık. Scaffold'un body bölümünde kendi widget'ımızı çağırdık.
- Burada ListView.builder yapılandırıcısı çağırılır.
- itemBuilder parametresine context ve index bilgisini verdik.
- ListView.builder metodu her satır için çalışacak.

```
@override
28 🌖
         Widget build(BuildContext context) {
        return Scaffold (
29
            — appBar: AppBar(
30

─ title: Text('Startup Adı Oluşturucu'),
             ), // AppBar
           — body: _buildSuggestions(),
34
           ); // Scaffold
36
       Widget _buildSuggestions() {
38
            return ListView.builder( // her satır için çalış ...
39
               padding: const EdgeInsets.all(16),
                itemBuilder: (BuildContext _context, int i) { //i satır index'ini tutar.
                 index O'dan başlar ve her çalıştırılmada artar.
42
43
                 Her sözcük eşleştirmesi için iki kez artar: bir kez ListTile için ve bir kez Divider() için.
                 Böylece, kullanıcı kaydırdıkça önerilen liste büyümeye devam eder
45
46
                 if (i.isOdd) { // eğer index tekse ( örn: 0-> ListTile , 1-> Divider , 2-> ListTile , 3-> Divider . . . )
48
                  – return Divider(); // Divider (çizgi) koy
49
50
                 final int index = i \sim / 2;
                 if (index >- suggestions length) { // index listeden uzunsa ( kullanıcı ekran kaudırıuorsa )
```

- Eğer index numarası tekse araya bir Divider yerleştirecek. Örneğin ilk başta 0 olduğu için bir ListTile (kelime) , daha sonra 1 'de Divider , ardından 2 'de tekrar Divider ve bu şekilde devam edecek.
- Eğer index , listemizdeki kelime sayısından (liste uzunluğu) büyük veya eşit olursa listeye 10 eleman daha ekleyecek. Bunu da paketteki metodlar sayesinde yapacak.
- Listedeki bu elemanları _buildRow metoduna vererek döndürecek.
- _buildRow kendi yazdığımız bir widget'tır. ListTile döndürür. Buradaki yazları PascalCase tipinde yazdırır. Font bilgisini ise sınıf içinde tanımladığımız _biggerFont değişkeninden alır.

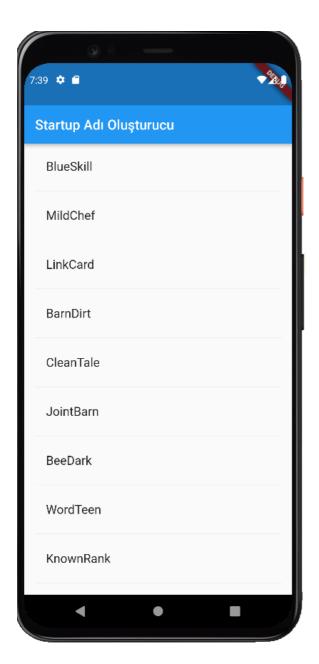
Ödev 4-2

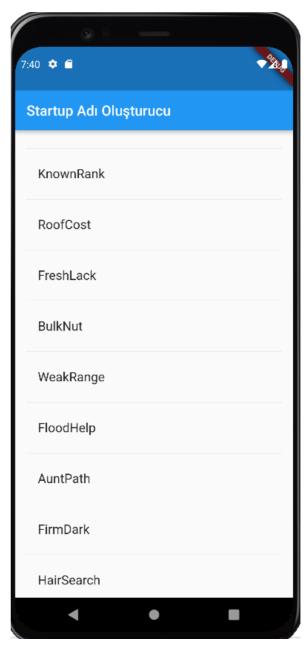
```
final int index = i \sim / 2;
51
52
53
54
                  if (index >= _suggestions.length) { // index , listeden uzunsa ( kullanıcı ekran kaydırıyorsa )
                   _suggestions.addAll(generateWordPairs().take(10)); // yenilerini ekle
             return _buildRow(_suggestions[index]);
 55
           ); // ListView.builder
 56
 57
 58
 59

    Widget _buildRow(WordPair pair) { // Satırlar oluşturulur.

 60
            return ListTile(
            ├ title: Text(
 61
          pair.asPascalCase,
 62
 63
                style: _biggerFont,
          ), // Text
 64
      ); // ListTile
 65
66
67
68
69
70
```

Ödev 4-2 4





Ödev 4-2 5