

Öğrenci Adı Soyadı : Furkan Kaçan

Öğrenci Numarası: 1030510300

Erciyes Üniversitesi Bilgisayar Mühendisliği Mobile Application Dersi Proje Ödevi

Dersin Hocası: Dr. Öğr. Üyesi Fehim Köylü

1. Proje Amacı

Bu projede, Flutter kullanarak DummyJSON API'sinden ürün verileri çekilecek ve ekranda bir liste halinde gösterilecektir.

Ayrıca, arama yaparak belirli ürünler bulunabilecektir.

2. Kullanılan Teknolojiler

- **Flutter**: Mobil uygulama geliştirme platformu
- **HTTP Paketi**: API'den veri çekmek için
- **DummyJSON API**: Ürün verilerini sağlayan sahte bir API

3. Genel Yapı

a) HTTP Paketi ile Veri Çekme

Flutter'da bir API'ye bağlanmak için http paketi kullanılır.

Örnek bir GET isteği:

```
import 'package:http/http.dart' as http;  
import 'dart:convert';
```

```
Future<void> fetchProducts() async {  
  final response = await http.get(Uri.parse('https://dummyjson.com/products'));  
  
  if (response.statusCode == 200) {  
    var data = jsonDecode(response.body);  
    print(data['products']); // Ürün verileri  
  } else {  
    print('Veri çekme başarısız');
```

```
}  
}
```

b) JSON Verilerini Modele Dönüştürme

API'den gelen ürün verilerini daha rahat kullanmak için bir Dart modeli tanımlanır:

```
class Product {  
  final String title;  
  final String description;  
  
  Product({required this.title, required this.description});  
  
  factory Product.fromJson(Map<String, dynamic> json) {  
    return Product(  
      title: json['title'],  
      description: json['description'],  
    );  
  }  
}
```

c) Flutter UI: Ürünleri Listeleme

Çekilen ürünler ekranda bir liste şeklinde gösterilir:

```
ListView.builder(  
  itemCount: products.length,  
  itemBuilder: (context, index) {  
    return ListTile(  
      title: Text(products[index].title),  
      subtitle: Text(products[index].description),  
    );  
  },  
)
```

```
);
```

4. Arama Özelliği

Kullanıcıların ürünler arasında arama yapabilmesi için bir arama kutusu eklenir:

```
TextField(  
  decoration: InputDecoration(  
    labelText: 'Ürün Ara',  
    suffixIcon: Icon(Icons.search),  
  ),  
  onSubmitted: (query) {  
    // Burada API'den arama sonuçları çekilebilir  
    print('Aranan kelime: $query');  
  },  
);
```

Arama için DummyJSON şu endpoint'i kullanır:

<https://dummyjson.com/products/search?q=iphone>

5. Özet

Bu proje sayesinde:

- API'den ürün verisi çekildi.
- Veriler liste halinde ekranda gösterildi.
- Kullanıcılar kategori seçerek veya arama yaparak ürünleri filtreleyebildi.

Bir Uygulama Örneği: `import 'package:flutter/material.dart';`

`import 'package:http/http.dart' as http;`

`import 'dart:convert';`

`void main() => runApp(const MyApp());`

`class MyApp extends StatelessWidget {`

```

const MyApp({super.key});

@override
Widget build(BuildContext context) {
  return MaterialApp(home: ProductListPage());
}

}

class Product {
  final String title;
  final String description;

  Product({required this.title, required this.description});

  factory Product.fromJson(Map<String, dynamic> json) {
    return Product(
      title: json['title'],
      description: json['description'],
    );
  }
}

class ProductListPage extends StatefulWidget {
  @override
  _ProductListPageState createState() => _ProductListPageState();
}

class _ProductListPageState extends State<ProductListPage> {
  late Future<List<Product>> productsFuture;

```

@override

```
void initState() {  
    super.initState();  
    productsFuture = fetchProducts();  
}
```

```
Future<List<Product>> fetchProducts() async {  
    final response = await http.get(Uri.parse('https://dummyjson.com/products'));  
    if (response.statusCode == 200) {  
        final data = jsonDecode(response.body);  
        List list = data['products'];  
        return list.map((e) => Product.fromJson(e)).toList();  
    } else {  
        throw Exception('Veri çekilemedi');  
    }  
}
```

@override

```
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(title: const Text('Ürün Listesi')),  
        body: FutureBuilder<List<Product>>(  
            future: productsFuture,  
            builder: (context, snapshot) {  
                if (snapshot.connectionState == ConnectionState.waiting) {  
                    return const Center(child: CircularProgressIndicator());  
                } else if (snapshot.hasError) {  
                    return Center(child: Text('Hata: ${snapshot.error}'));  
                } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
```

```
    return const Center(child: Text('Ürün bulunamadı'));  
  }
```

```
  final products = snapshot.data!;  
  return ListView.builder(  
    itemCount: products.length,  
    itemBuilder: (context, index) {  
      return ListTile(  
        title: Text(products[index].title),  
        subtitle: Text(products[index].description),  
      );  
    },  
  );  
},  
,  
,  
,  
);  
}  
}
```