

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. Öncelikle gerekli paketleri ekleyelim

pubspec.yaml dosyasına http paketini ekliyoruz.

dependencies:

flutter:

sdk: flutter

http: ^0.14.0

sonrasında terminalden flutter pub get komutunu çalıştıralım.

2. Model sınıfı oluşturalım.

```
class Product {  
  final int id;  
  final String title;  
  final String description;  
  final String thumbnail;
```

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

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

```
}  
}
```

3. API servisini yazalım.

```
import 'dart:convert';  
import 'package:http/http.dart' as http;  
import 'product.dart'; // model dosyanın ismi  
  
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 products = data['products'];  
    return products.map((e) => Product.fromJson(e)).toList();  
  } else {  
    throw Exception('Ürünler getirilemedi');  
  }  
}
```

4. UI – Listeyi gösterelim.

```
import 'package:flutter/material.dart';  
import 'product.dart';  
import 'api_service.dart';  
  
class ProductListPage extends StatelessWidget {  
  const ProductListPage({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: const Text('Ürün Listesi')),  
      body: FutureBuilder<List<Product>>(  
        future: fetchProducts(),  
        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('Hiç ürün bulunamadı.'));  
          }  
        }  
      )  
    );  
  }  
}
```

```

    }

    final products = snapshot.data!;
    return ListView.builder(
      itemCount: products.length,
      itemBuilder: (context, index) {
        final product = products[index];
        return ListTile(
          leading: Image.network(product.thumbnail, width: 50, height: 50, fit:
BoxFit.cover),
          title: Text(product.title),
          subtitle: Text(product.description),
        );
      },
    );
  },
),
);
}
}

```

5. main.dart dosyasına ekleyelim.

```

import 'package:flutter/material.dart';
import 'product_list_page.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'DummyJSON Ürünler',
      home: const ProductListPage(),
    );
  }
}

```

Kodun Anlatımı:

Proje Amacı

Bu proje, Flutter kullanarak DummyJSON API'sinden ürünleri çekip listelemek, kategorilere göre filtrelemek ve ürünlerde arama yapabilmek amacıyla geliştirilmiştir.

Kullanılan Araçlar ve Teknolojiler

- **Flutter:** Mobil uygulama geliştirme framework'ü
- **HTTP Paketi:** API'den veri çekmek için kullanıldı
- **DummyJSON API:** Ürün verilerini sağlayan sahte (dummy) API

Yapılan İşlemler

1. API ile Veri Çekme

- http paketi ile DummyJSON API'ye istek atıldı.
- Gelen JSON verileri Dart model sınıfına (Product) dönüştürüldü.

2. Ürün Listeleme

- Uygulamada FutureBuilder kullanılarak API'den çekilen ürünler ekranda ListView ile listelendi.
- Her ürün için:
 - Küçük bir resim (thumbnail)
 - Ürün başlığı (title)
 - Ürün açıklaması (description) gösterildi.

3. Kategoriye Göre Filtreleme

- API'den kategori isimleri çekildi (fetchCategories).
- Kullanıcı bir kategori seçtiğinde, o kategoriye ait ürünler gösterildi.

4. Arama Özelliği

- Üstte bir arama kutusu (TextField) eklendi.
- Kullanıcı ürün adı yazarak arama yapabiliyor.
- Arama sonrası ilgili ürünler listeleniyor.

Uygulamanın Akışı

1. Uygulama açılır → Kategoriler yüklenir.
2. Kullanıcı isterse kategori seçer → O kategoriye ait ürünler listelenir.
3. İsterse bir kelime ile arama yapar → Arama sonucuna göre ürünler listelenir.
4. Ürünler kaydırarak görüntülenebilir.

Önemli Kodlar

- Product Modeli: JSON veriyi Dart nesnesine çevirir.
- fetchProducts, fetchCategories, searchProducts: API isteklerini yapar.
- ProductListPage: UI katmanıdır. Listeyi ve arama kutusunu içerir.