

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROJE RAPORU



Öğrenci

İsim: Ömer Faruk

Soyisim: Akçal

Numara: 19290206

GITHUB:

<https://github.com/omerakcal33/Flutter-news>

Proje Özeti

Oluşturulan raporda dönem boyunca yapılan proje ve detayları ile ilgili bilgiler bulunmaktadır. Raporu yapılan projede SQL , flutter dilleri kullanılmış olup HTML , javascript,css dillerinden de faydalanarak front-end kısmı yapılmıştır. SQL kullanılarak database kısmında veriler tutulmuştur. Aynı zamanda proje gerçekleştirilirken visual studio ortamından yardım alınmıştır. Proje içerisinde tecrübe ettiğim bir başka konu ise API kavramının ne işe yaradığı ve nasıl kullanıldığı olmuştur.

Proje Hakkında Bilgiler

API (Application Programming Interface), yazılım uygulamaları arasında iletişim kurmak, veri alışverişi yapmak veya işlevsellik paylaşmak için kullanılan bir arayüzdür. Genel olarak, bir API, bir yazılımın başka bir yazılım tarafından kullanılmasını sağlar ve uygulamalar arasında veri ve işlevsellik paylaşımını kolaylaştırır. Uygulama entegrasyonu, veri alışverişi, işlevsellik paylaşımı gibi konularda işe yarar.

FRONT-END

LOGIN PAGE

Front-end kısmında yapılan işlemler javascript kullanılarak yazılmıştır. Arayüzün giriş sayfası şu şekildedir.



Login

- Username:
- Password:

Giriş sayfasının kodları aşağıdaki şekildedir.





Programın çıktısı yukarıda verilmiştir.

KODLAR

```
1 class ArticleModel {
2   String? status;
3   int? totalResults;
4   List<Articles>? articles;
5
6   ArticleModel(this.status, this.totalResults, this.articles);
7
8   ArticleModel.fromJson(Map<String, dynamic> json) {
9     status = json['status'];
10    totalResults = json['totalResults'];
11    if (json['articles'] != null) {
12      articles = <Articles>[];
13      json['articles'].forEach((v) {
14        articles!.add(Articles.fromJson(v));
15      });
16    }
17  }
18
19  Map<String, dynamic> toJson() {
20    final Map<String, dynamic> data = <String, dynamic>{};
21    data['status'] = status;
22    data['totalResults'] = totalResults;
23    if (articles != null) {
24      data['articles'] = articles!.map((v) => v.toJson()).toList();
25    }
26    return data;
27  }
28
29  class Articles {
30    String? status;
31    int? totalResults;
32    List<Articles>? articles;
33  }
```

```
C:\Users> cd C:\OneDrive> Masauzu > gereziflutter > news_app-main > ① README.md
1 class ArticleModel{
2   String? status;
3   int? totalResults;
4   List<Articles>? articles;
5
6
7   ArticleModel({this.status, this.totalResults, this.articles});
8
9   ArticleModel.fromJson(Map<String, dynamic> json) {
10     status = json['status'];
11     totalResults = json['totalResults'];
12     if (json['articles'] != null) {
13       articles = <Articles>[];
14       json['articles'].forEach((v) {
15         articles.add(Articles.fromJson(v));
16       });
17     }
18   }
19
20
21   Map<String, dynamic> toJson() {
22     final Map<String, dynamic> data = <String, dynamic>{};
23     data['status'] = status;
24     data['totalResults'] = totalResults;
25     if (articles != null) {
26       data['articles'] = articles!.map((v) => v.toJson()).toList();
27     }
28     return data;
29   }
30 }
31
32 class Articles {
33   String? source;
```

```
65   final Map<String, dynamic> data = <String, dynamic>{};
66   if (source != null) {
67     data['source'] = source!.toJson();
68   }
69   data['author'] = author;
70   data['title'] = title;
71   data['description'] = description;
72   data['url'] = url;
73   data['urlToImage'] = urlToImage;
74   data['publishedAt'] = publishedAt;
75   data['content'] = content;
76   return data;
77 }
78 }
79
80 class Source {
81   String? id;
82   String? name;
83
84   Source({this.id, this.name});
85
86   Source.fromJson(Map<String, dynamic> json) {
87     id = json['id'];
88     name = json['name'];
89   }
90
91   Map<String, dynamic> toJson() {
92     final Map<String, dynamic> data = <String, dynamic>{};
93     data['id'] = id;
94     data['name'] = name;
95     return data;
96   }
97 }
```

Articles kodları yukarıda verilmiştir.

KATEGORİLER VE KODLARI

```
3 List<CategoryModel> getCategories() {
4   List<CategoryModel> category = <CategoryModel>[];
5   CategoryModel categoryModel = new CategoryModel();
6
7   //1
8   categoryModel = new CategoryModel();
9   categoryModel.categoryText = "iş dünyası";
10  categoryModel.categoryName = "business";
11  categoryModel.imageUrl =
12    "https://images.unsplash.com/photo-1486406146926-c627a92ad1ab?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=870&q=80";
13  category.add(categoryModel);
14
15  //2
16  categoryModel = new CategoryModel();
17  categoryModel.categoryText = "spor";
18  categoryModel.categoryName = "sports";
19  categoryModel.imageUrl =
20    "https://images.unsplash.com/photo-1461896836934-ffe607ba8211?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1170&q=80";
21  category.add(categoryModel);
22
23  //3
24  categoryModel = new CategoryModel();
25  categoryModel.categoryText = "teknoloji";
26  categoryModel.categoryName = "technology";
27  categoryModel.imageUrl =
28    "https://images.unsplash.com/photo-1592659762303-90081d34b277?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1546&q=80";
29  category.add(categoryModel);
30
31  //4
32  categoryModel = new CategoryModel();
33  categoryModel.categoryText = "sağlık";
34  categoryModel.categoryName = "health";
35  categoryModel.imageUrl =
```

```
32  categoryModel = new CategoryModel();
33  categoryModel.categoryText = "sağlık";
34  categoryModel.categoryName = "health";
35  categoryModel.imageUrl =
36    "https://images.unsplash.com/photo-1587854692152-cbe660dbde88?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1738&q=80";
37  category.add(categoryModel);
38
39  //5
40  categoryModel = new CategoryModel();
41  categoryModel.categoryText = "eğlence";
42  categoryModel.categoryName = "entertainment";
43  categoryModel.imageUrl =
44    "https://images.unsplash.com/photo-1603739983239-8b6e64c3b185?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1742&q=80";
45  category.add(categoryModel);
46
47  //6
48  categoryModel = new CategoryModel();
49  categoryModel.categoryText = "bilim";
50  categoryModel.categoryName = "science";
51  categoryModel.imageUrl =
52    "https://images.unsplash.com/photo-1532094349884-543bc11b234d?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1170&q=80";
53  category.add(categoryModel);
54
55  //7
56  categoryModel = new CategoryModel();
57  categoryModel.categoryText = "genel";
58  categoryModel.categoryName = "general";
59  categoryModel.imageUrl =
60    "https://images.unsplash.com/photo-1495020689067-958852a7765e?ixlib=rb-1.2.1&ixid=MnwxtHjA3fDB8MhwaG90by1wYld1fhx8fGVufDB8fthx8&auto=format&fit=crop&w=1738&q=80";
61  category.add(categoryModel);
62
63  return category;
64 }
```

Bu kod, kategori modeli oluşturan ve bu kategorilerin bir listesini döndüren bir fonksiyondur. Fonksiyon, CategoryModel sınıfından nesneler oluşturur ve her bir kategoriye bu nesnelerle temsil eder. Bu kategorilerin her biri bir metin (categoryText), bir isim (categoryName) ve bir resim URL'si (imageUrl) içerir.

Bu kod, belirli kategorilerin metin, isim ve resim bilgilerini içeren bir liste döndürerek, genellikle haber uygulamalarında veya içerik gösteren uygulamalarda kategorileri göstermek için kullanılabilir.

API (APPLICATION PROGRAMMING INTERFACE)

API (Application Programming Interface), yazılım uygulamaları arasında iletişim kurmak, veri alışverişi yapmak veya işlevsellik paylaşmak için kullanılan bir arayüzdür. Flutter ile yapılan bir haber sitesi projesinde API'ler, haber verilerini dış kaynaklardan almak veya içeriği sunmak için kullanılabilir. Ayrıca API haber verilerinin alınması, kategori ve etiket bilgileri, güncelleme ve otomatik veri alımını sağlar. Bu projede son dakika verileri ile güncellemeyi haber sitesinde sağlamaktadır

BACK-END

```
1 def localProperties = new Properties()
2 def localPropertiesFile = rootProject.file('local.properties')
3 if (localPropertiesFile.exists()) {
4     localPropertiesFile.withReader('UTF-8') { reader ->
5         localProperties.load(reader)
6     }
7 }
8
9 def flutterRoot = localProperties.getProperty('flutter.sdk')
10 if (flutterRoot == null) {
11     throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")
12 }
13
14 def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
15 if (flutterVersionCode == null) {
16     flutterVersionCode = '1'
17 }
18
19 def flutterVersionName = localProperties.getProperty('flutter.versionName')
20 if (flutterVersionName == null) {
21     flutterVersionName = '1.0'
22 }
23
24 apply plugin: 'com.android.application'
25 apply plugin: 'kotlin-android'
26 apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
27
28 android {
29     compileSdkVersion flutter.compileSdkVersion
30     ndkVersion flutter.ndkVersion
31
32     compileOptions {
33         sourceCompatibility JavaVersion.VERSION_1_8
34         targetCompatibility JavaVersion.VERSION_1_8
35     }
36 }
```

DATABASE –SQL

SQL (Structured Query Language), ilişkisel veritabanı yönetim sistemlerinde (RDBMS) veritabanları oluşturmak, değiştirmek, sorgulamak ve yönetmek için kullanılan bir programlama dilidir. SQL, veritabanlarıyla etkileşim kurmak için kullanılır ve birçok farklı RDBMS tarafından desteklenir.

Bu projede SQL kullanılma amacı veri ekleme, güncelleme, silme ayrıca veri ilişkilendirmeleri ve tablo yapıları için kullanılmıştır.