



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**English For Kids**  
**MAHAMAT KABIR SOULEYMAN ISSEIN**  
**TASARIM ÇALIŞMASI II**

**DANIŞMAN: Dr. Öğr. Üyesi Nihan KAZAK ÇERÇEVİK**

**BİLECİK**  
**23 Haziran 2022**



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**English For kids**  
**MAHAMAT KABIR SOULEYMAN ISSEIN**  
**TASARIM ÇALIŞMASI II**

**DANIŞMAN: Dr. Öğr. Üyesi Nihan KAZAK ÇERÇEVİK**

**BİLECİK**  
**23 Haziran 2022**

## **ÖZET**

### **Projenin Amacı**

Bu proje, çocuklara basit İngilizce kelimelerin temellerini öğrenmenin kolay bir yolunu sunarken, oyunlar gibi etkili araçlar yaparak öğrendiklerini hatırlamalarını ve İngilizce seviyelerini geliştirmelerini sağlamayı amaçlamaktadır.

### **Projenin Kapsamı**

Bu projenin kapsamı, kullanıcının bir kez ana sayfaya gelip, sayılar, alfabeler, renkler gibi çeşitli bölümlerde listelenen bazı temel İngilizce bilgilerini öğrenmesi için oluşturulmuştur. Bu bölümden sonra ses, görüntü ve bazı pekiştirme alıştırmaları (kelime bulma, hareketli öğeler vb.) ile öğrenme sayfası gelmektedir. Ayrıca kullanıcı kendi diliyle İngilizce kelimeleri karşılayıp kolayca öğrenmesi sağlamaktadır.

### **Sonuçlar**

Bu uygulamayı kullanan herhangi bir kullanıcı İngilizce'nin temellerini öğrenecek hale gelebilir ayrıca kolayca kelimeler okuyabilir, yazabilir ve anlayabilir.

## **ABSTRACT**

### **Project Objective**

This project aims to provide children with an easy way to learn the basics of simple English words, while helping them to remember what they have learned and improve their English level by making effective tools such as games.

### **Scope of Project**

The scope of this project is created for the user to come to the home page once and learn some basic English knowledge listed in various sections such as numbers, alphabets, colors. After this section, learning page with audio, image and some reinforcement exercises (word finding, moving items etc.) In addition, the user can meet and learn English words in his own language easily.

### **Results**

Any user using this application will become able to learn the basics of English and can easily read, write and understand words.

## **TEŞEKKÜR**

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve beni bu konuya yönlendiren saygıdeğer hocam ve danışmanım Sayın Dr. Öğr. Üyesi Nihan KAZAK ÇERÇEVİK 'a tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

**Mahamat Kabir SOULEYMAN ISSEIN**

23 Haziran 2022

# İÇİNDEKİLER

ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGE LİSTESİ	vii
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	ix
<b>1 GİRİŞ</b>	<b>1</b>
<b>2 PROJEDE KULLANILAN YAZILIMLAR VE YÖNTEMLER</b>	<b>2</b>
2.1 Flutter . . . . .	2
2.1.1 Widget Kavramı . . . . .	2
2.1.2 Stateless widget . . . . .	2
2.1.3 Statefull widget . . . . .	3
2.2 Visual Studio Code . . . . .	3
2.3 Flutter Audio Player Paketi . . . . .	4
2.3.1 AudioCache() . . . . .	6
2.3.2 AudioPlay() . . . . .	6
2.4 Flutter Localization Paketi . . . . .	7
2.5 Font Awesome Paketi . . . . .	12
2.6 Animated Splash Screen Paketi . . . . .	12
2.7 Pubspec.yaml Dosyası . . . . .	12
<b>3 PROJENİN TASARIMI VE ÇALIŞMA YAPISI</b>	<b>13</b>
3.1 Yeni Flutter Project Oluşturma . . . . .	13
3.2 Açılış Ekranın Özelleştirme . . . . .	14
3.3 Projenin Tasarımı . . . . .	15
3.3.1 Ana Sayfa Menu . . . . .	15

3.4	Bölüm Seçimi . . . . .	18
3.4.1	Öğrenme Kısmı . . . . .	18
3.4.2	Kelime Bulma . . . . .	19
3.4.3	sürükle bırak . . . . .	19
3.4.4	Yazma . . . . .	20
<b>4</b>	<b>SONUÇLAR VE ÖNERİLER</b>	<b>21</b>
<b>5</b>	<b>EKLER</b>	<b>22</b>
	<b>KAYNAKLAR</b>	<b>23</b>
	<b>ÖZGEÇMİŞ</b>	<b>24</b>

## SİMGE LİSTESİ



## ŞEKİL LİSTESİ

1	Audio klasör . . . . .	4
2	Audio Asset . . . . .	5
3	Localization dependencies . . . . .	7
4	Localization dependencies . . . . .	8
5	Assets Localization dependencies . . . . .	8
6	SupportedLocales . . . . .	9
7	LocalizationDelegates . . . . .	9
8	Diller List . . . . .	11
9	SplashScreen . . . . .	15
10	json sonucu . . . . .	16
11	Ana Menu . . . . .	16
12	Seçim bölüm . . . . .	18
13	Öğrenme Kismi . . . . .	18
14	Kelime Bulma . . . . .	19
15	sürekleme . . . . .	20
16	Yazma . . . . .	20

## TABLO LİSTESİ

# 1 GİRİŞ

İngilizce, günümüzde hızla dünyanın en yaygın kullanılan ve konuşulan dili haline geldi ve bu büyümeyle birlikte çocuklar için İngilizce öğrenme ihtiyacı katlanarak arttı. İngilizce öğrenmek, yeni beceriler edinmenin harika bir yoludur ve her yaşta yapılabilir. Bununla birlikte, bunu çocuklukta öğrenmenin uzun vadeli birçok faydası vardır. Bu yüzden onu İngilizce'ye alıştırmak için iyi bir zaman. duyacağı yeni İngilizce kelimeler uzun süre hafızasında kazanmış olarak kalacaktır. bir dille tanışmak için ideal yaş aralığı çocukluk-tur. Bu aşamada yeni kavramları anlama yeteneği doğal olarak gelir. Bu nedenle çocuklar İngilizceyi daha hızlı öğrenme eğilimindedir. Bununla birlikte, başka birçok avantajı da vardır.

Bu koleksiyon, çocuğunuzun oyun oynarken ve eğlenirken önemli analitik ve bilişsel becerilerini geliştirmesini ve geliştirmesini sağlar. Bu uygulamalar, çocuğunuzun sınıf ortamı dışında öğrenmede başarılı olmasına yardımcı olmak için sınıfta atılan temeller üzerine kuruludur. Uygulama, harf seslerini öğretmeye, karıştırmaya ve kod çözmeye odaklanır. Oyunlar bağımsız öğrenmeyi teşvik etmek için tasarlanmıştır.

Çocukların dili ve sesleri taklit etme konusunda büyük bir kapasiteleri vardır. Son derece spontanedirler ve kendilerini yaşlı hissetmenin gülünç olduğu konusunda hiçbir fikirleri yoktur, bu da onları ne zaman İngilizce konuşacakları ve pratik yapacakları konusunda kısıtlanmış hissetmelerine neden olmaz. Böylece çocuklar taklit yoluyla sesleri ve cümleleri çalar ve ikinci bir dili daha hızlı öğrenebilirler.

Sonuç olarak, günümüzde İngilizce iletişim kurmanın öneminin daha çok farkına varmalı ve ebeveynler, çocuğun bir an önce öğrenmeye başlaması için her türlü çabayı göstermelidir.

## 2 PROJEDE KULLANILAN YAZILIMLAR VE YÖNTEMLER

### 2.1 Flutter

Bu projede Google tarafından geliştirilen Flutter kullanıldı. Flutter, Google'ın mobil, masaüstü ve web uygulamaları geliştirmek için sunmuş olduğu açık kaynak kodlu bir yazılım geliştirme aracıdır. Flutter uygulamaları yine Google tarafından geliştirilen ve nesne tabanlı bir dil olan Dart programlama dili ile yazılmaktadır. Aşağıdaki avantajları sebebiyle Flutter tercih edildi. [1]

- Android ve iOS uygulamaları için aynı kod kullanıldığından geliştirme hızı yüksektir. Flutter tekil kod tabanına sahiptir. Bu durum uygulama geliştirme maliyetini olabildiğince düşürür. Ayrıca uygulamanın pazara sunulma süresini hızlandırır.
- Flutter gerçek zamanlı uygulamalara eşit performans sunar. Diğer yazılım geliştirme yapılarına göre uygulamaların yerel bileşenlerle etkileşim kurması zorunlu değildir. Söz konusu etkileşim köprüler aracılığıyla gerçekleştirilir ve bunlar performans sorunlarına yol açabilirler.
- Flutter kullanan geliştiriciler yapılan tüm değişiklikleri gerçek zamanlı olarak görme fırsatına sahiptirler. Uygulama çalışırken değiştirilen kod hemen yansır ve yeniden başlatma gerekli değildir. Hata düzeltme ve özellik eklemek oldukça basittir.

#### 2.1.1 Widget Kavramı

Flutter'ın sloganı 'Everything widgettır'. Bu ifadeden de anlaşılacağı üzere widget Flutter uygulamaları için çok önemlidir. Main fonksiyonu içerisinde runApp fonksiyonu da bir widget döndürür. Her ekranın ana iskeleti(scaffold) widgettır. Flutter'da iki tür widget vardır : Stateless Widget ve Stateful Widget.

#### 2.1.2 Stateless widget

Stateless widgetta kod tekrar çalışmadığı veya ekranı sıfırdan oluşturmadığımız sürece ekranda bir değişiklik görülemez. Ekran bir kere çizilir ve öyle kalır. Ana ekran bir kere

çağırılır. Bir kere çizilir. Böylelikle uygulama daha performanslı çalışır. Aşağıdaki kodundan oluşmaktadır

```
1 class HommePage extends StatelessWidget {  
2   const HommePage({ Key? key }) : super(key: key);  
}
```

### 2.1.3 Statefull widget

Widget içerisinde herhangi bir değişken değişimi veya ekranın yeniden çizilmesi durumları olduğunda kullanılır. İhtiyaç duyulan bölümde setState fonksiyonu çağırılır. Build fonksiyonunun tekrar çalışmasıyla ekran yeniden çizilir. Uygulamada gerekli görülen yerlerde Stateful widget kullanılmıştır. Performanslı çalışma için olabildiğinde Stateful widget kullanımından kaçınılmıştır. Aşağıdaki kodundan oluşmaktadır.

```
1 class TextFieldWidget extends StatefulWidget {  
2   State<TextFieldWidget> createState() => _TextFieldWidgetState();  
3 }
```

## 2.2 Visual Studio Code

Visual Studio Code, Windows, macOS ve Linux'te çalışabilen, ücretsiz, açık kaynak ve hafif fakat güçlü bir kaynak kodu editörüdür. Javascript, TypeScript ve Node.js ile birlikte C++, C#, Python, PHP, Go gibi diller için zengin eklenti desteğine sahiptir. Aşağıdaki avantajları sebebiyle Visual Studio Code tercih edildi.

- Zengin bir kod refactoring desteği vardır.
- Platform bağımsız kod geliştirebilmeyi destekler.
- Zengin bir kod tanıma(code intellisense) içeriği vardır.
- Code Debugger desteği vardır.
- Hızlıdır.

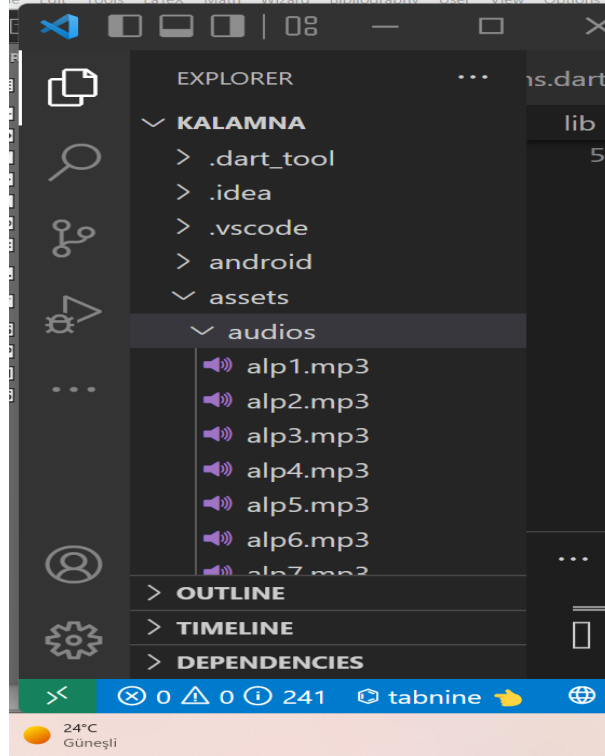
## Paketler

Açık kaynak paketleri, özellikle Flutter uygulamaları oluşturmaya gelince, yazılım geliştirmede büyük rol oynar. Pub.dev, Flutter uygulama işlevsellik eklemesi ve topluluk tarafından oluşturulan çözümlerden yararlanması yardımcı olmak için 24.000'den fazla paketten oluşan büyüyen bir ekosisteme sahiptir. Bu projede, kullanmış olduğum Flutter ve Dart paketlerinin derlenmiş bir listesini içermektedir.

### 2.3 Flutter Audio Player Paketi

Hem iOS hem de Android mobil platformlarını destekleyen audioplayers paketini kullanarak Flutter'da sesi uygulanabilir. Paket web platformları ile uyumludur. Bu projede sizlere Flutter'a nasıl ses oynatacağını göstermiş (kod örnekleri ile) olacak.

**Audio player yapısı:** Bu görüntü, uygulamadaki çarpıntı ses çalar entegrasyonu için proje yapısını sağlar.[2]

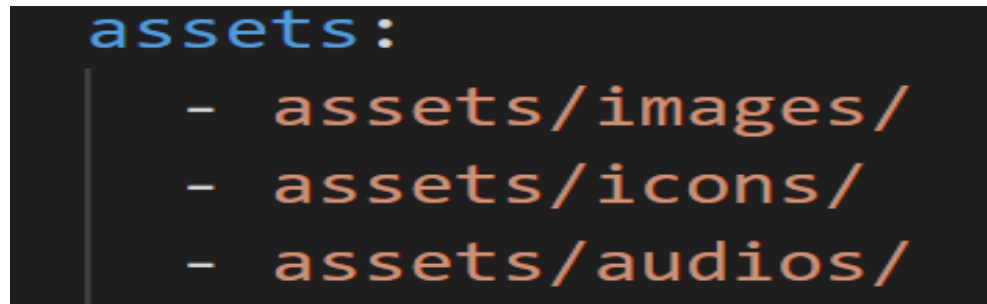


Şekil 1: Audio klasör

ilk yapılacak şey , aşağıdaki paketleri [pubspec.yaml](#)'e eklemek ve ardından [flutter pub get](#)'i çalıştırmaktır.

```
1 dependencies: \linebreak
2   audioplayers:0.19.0
3   http: 0.13.3
4   pathprovider: 2.0.2
```

Flutter müzik çalarda sese erişmek ve ses çalmak için pubspec'e varlıklar eklenecek. Aşağıdaki şekil:2 gibi



```
assets:
  - assets/images/
  - assets/icons/
  - assets/audios/
```

Şekil 2: Audio Asset

**iOS:** Info.plist dosyasını aşağıdaki kodla düzenlenecek

```
1 <key>NSAppTransportSecurity </key>
2 <dict>
3   <key>NSAllowsArbitraryLoads </key>
4   <true/>
5 </dict>
```

**Android:** İçinde bulunan [AndroidManifest.xml](#) dosyasını aşağıdaki kodla düzenlenecek

android/app/src/main/AndroidManifest.xml

```
1 <manifest ...>
2   <uses-permission android:name="android.permission.INTERNET" />
3   <application ...
4     android:usesCleartextTraffic="true" ...>
5     ...
6   </application>
7 </manifest>
```

**Widget oluşturma:** Şimdi sesi işleyecek Widget'ı oluşturalacak; Her şeyden önce, paketi widget'ı aktarmak:

```
import 'package:audioplayers/audioplayers.dart';
```

### 2.3.1 AudioCache()

Local Assets oynamak için AudioCache sınıfını kullanacak. Flutter, varlıklarında ses çalmak için kolay bir yol sağlamaz, ancak bu sınıf çok yardımcı olur. Aslında varlığı cihazdaki geçici bir klasöre kopyalar ve burada daha sonra Yerel Dosya olarak oynatılır. Bir önbellek görevi görür, çünkü kopyalanan dosyaları izler, böylece gecikmeden yeniden oynatabilir; ayrıca dosyaları önceden yüklenmeli. Her örneğin kendi bağımsız önbelleği vardır, bu nedenle yükleme süreleri optimize etmek için bir singleton oluşturmuştur. Bir ses çalmak için şunu çalıştırılacak:

```
static AudioCache player = new AudioCache();  
player.play('explosion.mp3');
```

Bu, proje varlık klasördeki [audio](#) dosyasını oynatır.

### 2.3.2 AudioPlay()

halihazırda çalınmakta olan ses dosyasını duraklatmak veya devam ettirmek için kullanılır.[6]

```
1 class _AlphabetScreen1State extends State<AlphabetScreen1> {  
2   AudioCache audioCache = AudioCache();  
3   AudioPlayer audioPlayer = AudioPlayer();  
4   @override  
5   void dispose() {  
6     audioPlayer.stop();  
7     super.dispose();  
8   }
```



## 2.4 Flutter Localization Paketi

Bir uygulama oluşturuyorsak ve uygulamanın kullanımını artırmak istiyorsak, o zaman uygulamada birden çok dili desteklememiz gerekiyor. Yerelleştirme yardımı ile uygulamanın dilini değiştirebilirim çünkü Android ve iOS en popüler olanlarıdır. Bu nedenle, her iki tür cihazda ve her iki alanda da konumu görüntülemek için çarpıntı konumu kullanıyorum.[3]

### Birden çok dilin bazı Temel gereksinimleri.:

- Flutter Localization uygulaması, varsayılan olarak akıllı telefonda yapılandırılan dile göre çalışır.
- Uygulamada bir dil desteklenmiyorsa, İngilizce (tr) varsayılan dildir.
- Son kullanıcı, desteklenen diller listesinden çalışma dilini değiştirebilir.
- Kullanıcı diğer dillerden herhangi birini seçtiğinde, seçilen dilin değerlerini görüntülemek için uygulamanın tüm düzeni yenilenir.

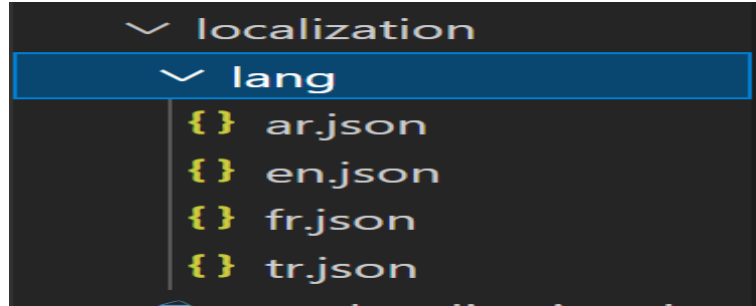
### Kullanım:

İlk adım, pubspec.yaml dosyasına localization'ı bir bağımlılık olarak eklemektir.[7]

```
32 flutter:  
33   sdk: flutter  
34 flutter_localizations:  
35   sdk: flutter  
36 shared_preferences: ^2.0.15
```

Şekil 3: Localization dependencies

Bir klasör oluşturulacak ve bunun gibi çeviri dosyaları eklenmeli.



Şekil 4: Localization dependencies

Localization assets dizini pubspec.yaml’de bildirinecek:

```
assets:
  - assets/images/
  - assets/icons/
  - assets/audios/
  - assets/shapes/
  - lib/localization/lang/gr.json
  - lib/localization/lang/ar.json
  - lib/localization/lang/tr.json
  - lib/localization/lang/fr.json
```

Şekil 5: Assets Localization dependencies

iOS ile ilgili not Çevirinin iOS üzerinde çalışması için ios/Runner/Info.plist’e desteklenen yerel ayarlar eklemek gerekir.

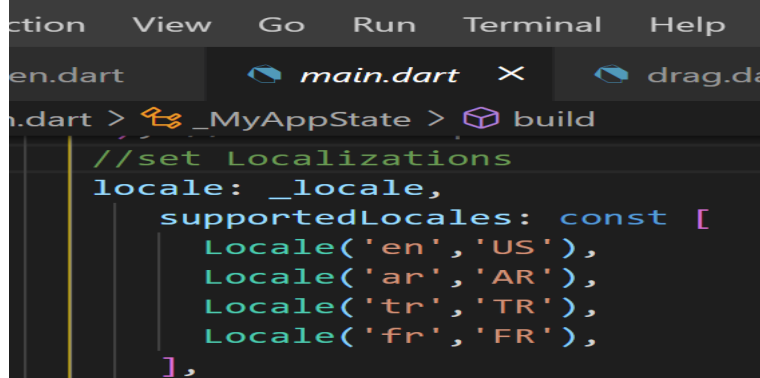
Örnek:

```
1 <key>UIViewControllerBasedStatusBarAppearance </key>
2   <key>CFBundleLocalizations </key>
3   <array>
4     <string>ar </string>
5     <string>en </string>
6     <string>fr </string>
7     <string>tr </string>
8   </array>
```

MaterialApp için yerelleştirme ekler:

Uygulamada localization uygulamak için, MyApp() sınıfının içine MaterialApp widget supportedLocales ve localizationsDelegates eklenecek.

**supportedLocales Ekleme:** Bu, varsayılan olarak ABD İngilizcesini tanıdığı için yerel olarak sağladığı tüm dilleri ayarlamak için kullanılır. aşağıdaki şekil 6 da gösterilmiştir.

A screenshot of an IDE window showing a Dart file named 'main.dart'. The code is inside a widget's 'build' method, specifically within a 'locale' property. It defines a 'supportedLocales' list with four entries: 'en' (US), 'ar' (AR), 'tr' (TR), and 'fr' (FR).

```

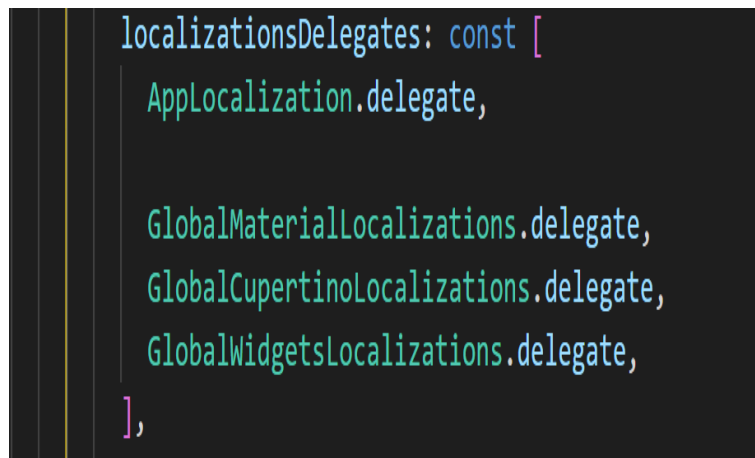
//set Localizations
locale: _locale,
supportedLocales: const [
  Locale('en', 'US'),
  Locale('ar', 'AR'),
  Locale('tr', 'TR'),
  Locale('fr', 'FR'),
],

```

Şekil 6: SupportedLocales

localizationsDelegates Material App ekleme:

- **GlobalMaterialLocalizations.delegate** donanım bileşenleri için yerelleştirilmiş dizeleri ve diğer değerleri almak için kullanılır.
- **localizationsDelegates** yerelleştirme delegelerinden yerelleştirilmiş değerlerin toplanması için MaterialApp'da listelenen bir özelliktir.
- **GlobalWidgetsLocalizations.delegate** widget kitaplığı için varsayılan metin yönünü tanımlamak için kullanılır. Aşağıdaki Şekil 8 da gösterilen bir örnektir.

A screenshot of an IDE window showing a Dart file. The code defines a 'localizationsDelegates' list containing four delegates: 'AppLocalization.delegate', 'GlobalMaterialLocalizations.delegate', 'GlobalCupertinoLocalizations.delegate', and 'GlobalWidgetsLocalizations.delegate'.

```

localizationsDelegates: const [
  AppLocalization.delegate,

  GlobalMaterialLocalizations.delegate,
  GlobalCupertinoLocalizations.delegate,
  GlobalWidgetsLocalizations.delegate,
],

```

Şekil 7: LocalizationDelegates

Artık uygulamanız dili değiştirmeye hazır. TranslationDemo sınıfını durum bilgisi olan bir pencere ögesi, istenilen pencere ögesi ve tercih edilen bir kullanıcı arabirimi tasarımı olarak oluşturabilir. Kendi dilinde JSON dosyaları, çevirmeniz gereken kelimeleri ekleyecek. İstenilen kadar JSON dosyası eklenip ve dili desteklenen yerel ayarlara eklenebilir. Demo için kullanmış olduğu anahtarlar aşağıdaki gibidir.

```
1  "number_one" : "daya",
2  "number_two": "biyu",
3  "number_tree" : "itace",
4  "number_four" : "hudu",
5  "number_five" : "biyar",
6  "number_six": "shida",
7  "number_seven" : "bakwai",
8  "number_eight": "takwas",
9  "number_nine": "na tara",
10 "product_writing": "rubuta",
11 "product_drop": "Drop",
12 "product_find": "Nemi Kalma",
```

Bundan sonra, içinde widget metnimizde değeri belirleyeceğimiz bir açılır buton kullandığımız FlutterLocalizationDemo adında bir sınıf oluşturacağız ve dilin hangi dilde kullanılacağını belirleyeceğimiz title adında bir fonksiyon oluşturduk. kodu, Ülke kodu vb. herhangi bir şey olabilir, açılır menüden dili seçtiğimiz anda, verilen başlığı da değiştirecektir.

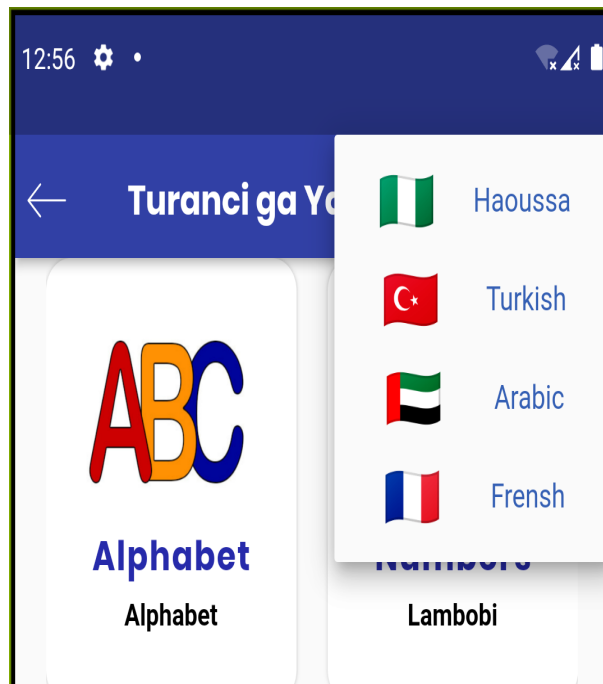
```
1  actions: <Widget>[
2      Padding(
3          padding: const EdgeInsets.all(8.0),
4          child: DropdownButton<Language>(
5              underline: const SizedBox(),
6              icon: const Icon(
7                  Icons.language,
8                  color: Color.fromARGB(255, 255, 255, 255),
9                  size: 40.0,
10             ),
11             onChanged: (language) {
12                 _changeLanguage(language!);
13             },
```

```

14         items: Language.languageList()
15         .map<DropdownMenuItem<Language>>(
16             (e) => DropdownMenuItem<Language>(
17                 value: e,
18                 child: Row(
19                     mainAxisAlignment:
20                     MainAxisAlignment.spaceAround,
21                     children: <Widget>[
22                         Text(
23                             e.flag,
24                             style: const TextStyle(fontSize: 30),
25                         ),
26                         Text(e.name)
27                     ],
28                 ),
29             ).toList(),
30     ),
31 ),
32 ],

```

Uygulamayı çalıştırdığında alttaki ekran görüntüsü gibi ekranın çıktısını almalıyız.



Şekil 8: Diller List

## 2.5 Font Awesome Paketi

İçerisinde ikonlar bulunan aynı zamanda ikon tasarlanan flutter paketidir. Bu projede testler bölümü yazılırken konu başlığı üzerinde bulunan ikonlar kısmında kullanılmıştır.[5]

## 2.6 Animated Splash Screen Paketi

Bu paket ile animasyonlu giriş ekranı yapıldı. Kullanıcı sadece giriş yap ekranından önce bu ekranla karşılaşacaktır. Flutter uygulamayı ilk açtığında default olarak kısa süreli de olsa beyaz bir ekran getirir. Bu ekranın olmaması ve uygulama başlangıcının daha renkli olması için bu paket kullanıldı.[4]

## 2.7 Pubspec.yaml Dosyası

Bu dosya projeye özgü ayarları içeren dosyadır. Uygulama ismi, pub.dev’de yayınlanıp yayınlanmayacağı, versiyonu, yazım tarihinde kullanılan sürümü görülebilir. Uygulamada kullanılan hazır paketler; resim, video gibi dosyaların tutulduğu bölümleridir. Aşağıdaki Kodu gösterilmiştir.

```
1 name: kalamna
2 description: A new Flutter project.
3 publish_to: 'none' # Remove this line if you wish to publish to pub.dev
4 version: 1.0.0+1
5 environment:
6   sdk: ">=2.16.2 <3.0.0"
7 dependencies:
8   provider: ^6.0.3
9   audioplayers: ^0.20.1
10  flutter:
11    sdk: flutter
12  flutter_localizations:
13    sdk: flutter
14  shared_preferences: ^2.0.15
15  cupertino_icons: ^1.0.2
16  font_awesome_flutter: ^10.1.0
17  google_fonts: ^2.1.0
18  flutter_svg: ^1.0.3 #it help to us svg in our app
19  flutter_bloc: ^8.0.1
20  rflutter_alert: ^2.0.4
```

```
21  rxdart: ^0.27.3
22  get: ^4.6.1
23  animated_splash_screen:
24  curved_navigation_bar: ^1.0.3
25  webview_flutter: ^3.0.4
26 dev_dependencies:
27   flutter_launcher_icons: "^0.9.2"
28   flutter_test:
29     sdk: flutter
30   flutter_lints: ^1.0.0
31 flutter_icons:
32   android: "launcher_icon"
33   ios: true
```

## 3 PROJENİN TASARIMI VE ÇALIŞMMA YAPISI

### 3.1 Yeni Flutter Project Oluşturma

Öncelikle yeni bir Flutter projesi oluşturdum. Bunun için Flutter SDK ve Flutter uygulama geliştirme ile ilgili diğer bileşenlerin kurulu olduğundan emin olmak lazım. Her şey düzgün bir şekilde ayarlandıktan sonra, bir proje oluşturmak için aşağıdaki komutu yerel dizinde çalıştırmak yeterlidir:

```
flutter create kalamna
```

Proje kurulduktan sonra, proje dizini içinde gezinin ve projeyi mevcut bir emülatörde veya gerçek bir cihazda çalıştırmak için terminalde aşağıdaki komutu yürütmektedir:

```
flutter run
```

Uygulamanın yapımına başlarken ilk olarak pubspec.yaml dosyasını açıp ve aşağıdaki örnekte gösterildiği gibi splashscreen paketini eklemiştik.

```
dependencies: flutter: sdk: flutter splashscreen: 1.3.5
```

## 3.2 Açılış Ekranın Özelleştirme

Uygulamanın açılış ekranın üzerinde metin ve görselle görüntülemesini için, oldukça özelleştirilebilir olan SplashScreen widget'ın kullanmıştır.Önce aşağıdaki kodlara bakarak ve ardından ne işe yaradıklarını öğrenmiş olacak.Aşağıdaki kod gibi gösterilmiştir.

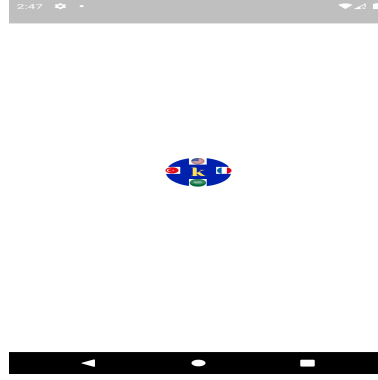
```
1
2 return MaterialApp(
3     debugShowCheckedModeBanner: false ,
4     title: 'Flutter Demo' ,
5     theme: ThemeData(
6         textTheme: Theme.of(context).textTheme.apply(bodyColor:
7             kTextColor) ,
8         primarySwatch: Colors.blue ,
9     ),
10    home: AnimatedSplashScreen(
11        splash: Image.asset(
12            'assets/images/kalam1.png' ,
13            height: 140.0 ,
14            width: 110.0 ,
15            fit: BoxFit.fitHeight ,
16        ),
17        duration: 500 ,
18        splashTransition: SplashScreen.rotationTransition ,
19        backgroundColor: const Color.fromARGB(255, 255, 255, 255) ,
20        nextScreen: WelcomePage() ,
21    ),
```

Yukarıda, aşağıdaki özelliklere sahip bir SplashScreen widget'ı kullanan basit bir uygulama oluşturmuştur.

- **seconds:** Açılış ekranın kaç saniye gösterileceği
- **navigateAfterSeconds:** Açılış ekranından sonra yönlendirmek istediğiniz widget
- **title:** Açılış ekranında göstermek istenilen başlık
- **style:** Başlığın veya açılış ekranındaki herhangi bir metnin stilini ayarlamak için
- **image:** Açılış ekranında göstermek istenilen görsel



Yukarıdaki kod örneğini bir emülatörde çalıştırırsa, uygulama açılış ekranı aşağıdaki gör- selde gösterilen gibi görünecektir.



Şekil 9: SplashScreen

### 3.3 Projenin Tasarımı

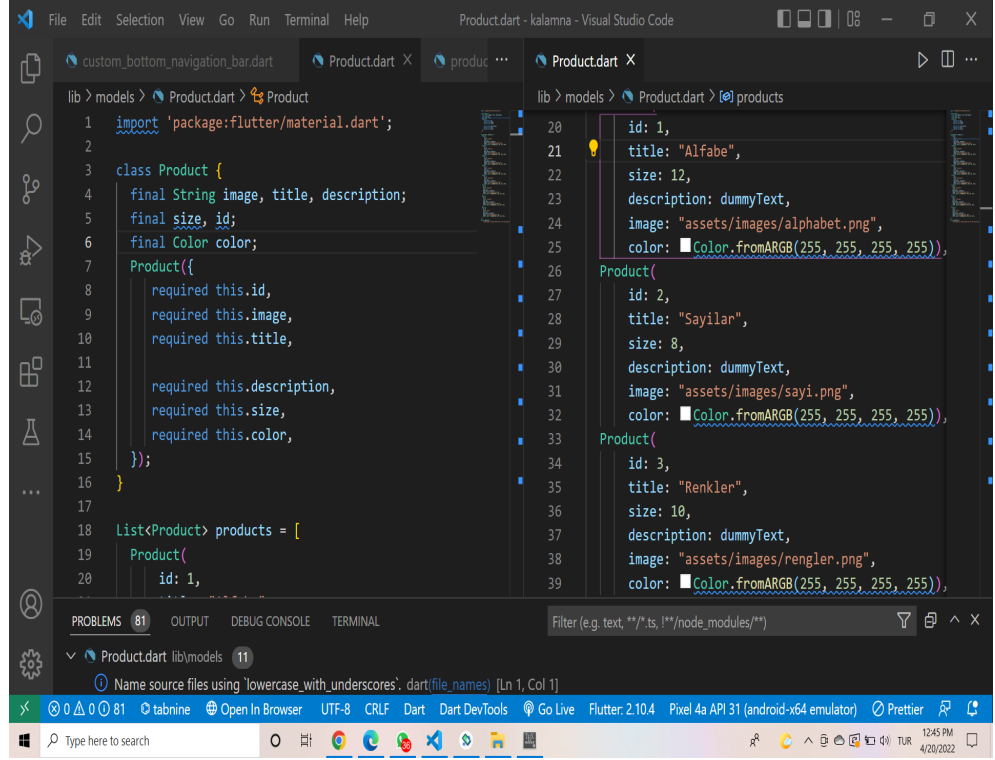
Bu uygulama çocuklar için tasarlandı, bu yüzden yapımı anlaşılır ve kullanımı kolay hale getirmek için elimden gelenin en iyisini yaptım. Menü, çocuğun seçebileceği ve kendi se- çimiyle öğrenmeye çalışabileceği tüm öğeleri içerir. alfabeyi, sayıları, renkleri, hayvanları bulabileceğimiz...

#### 3.3.1 Ana Sayfa Menu

son yazıda Flutter ile proje kurulumu bahsetmişti.Bu yazıda, adım adım, Flutter ana ekranı aşağıdaki Özellikle oluşturmuştur:

- Temel Flutter widget'larını kullanarak kullanıcı arabirimi oluşturma
- özel widget ve animasyonlar oluşturma
- Flutter'da varlıkları kullanma
- kolayca bakımı yapılabilecek kodu yazmak.
- **image:** Açılış ekranında göstermek istediğiniz görsel

Sunucundan gelecek olan cevapta kelime adı ,resim ve renk içeren bir JSON objesi dö- necek.Kullanıcı listesini seri durumdan çıkarmak için bir User sınıfına ihtiyac var:



Şekil 10: json sonucu

body yönteminin girişinde Json statik yöntemi aracılığıyla çağrılır ve kelime durumu ayarlanır . Bu durum ayarı, derleme yöntemini çalışırsak böyle bir sonuç çıkmış olacak.Aşağıdaki şekil 11 gösterilmiştir.



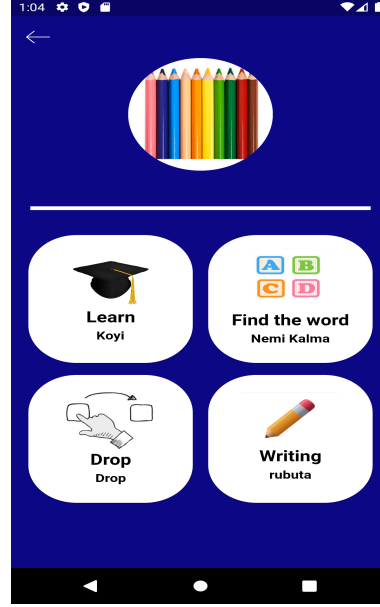
Şekil 11: Ana Menu

## Ana menu kod kısmı

```
1 class Body extends StatelessWidget {
2
3   @override
4   Widget build(BuildContext context) {
5     return Column(
6       crossAxisAlignment: CrossAxisAlignment.start,
7       children: <Widget>[
8         Expanded(
9           child: Padding(
10            padding: const EdgeInsets.symmetric(horizontal:
11              kDefaultPaddin),
12            child: GridView.builder(
13              itemCount: products.length,
14              gridDelegate: const
15                SliverGridDelegateWithFixedCrossAxisCount(
16                  crossAxisCount: 2,
17                  mainAxisSpacing: kDefaultPaddin,
18                  crossAxisSpacing: kDefaultPaddin,
19                  childAspectRatio: 0.75,
20                ),
21              itemBuilder: (context, index)=> ItemCard(
22                product: products[index],
23                press: () => Navigator.push(
24                  context,
25                  MaterialPageRoute(
26                    builder: (context) => DetailsScreen(
27                      product: products[index],
28                    )
29                )), ) ),
30       ],
31     }
```

### 3.4 Bölüm Seçimi

bu kısım kullanıcı öğrenmek istediği kısmı seçtikten hemen sonra gelir. birkaç farklı bölüme ayrılmıştır[11]



Şekil 12: Seçim bölüm

#### 3.4.1 Öğrenme Kismi

Bu bölümde çocuklar listelen kelimeler , istenilen birini üzerine tıklayarak ingilizcesi hem dinleyebilir ve yazılı olarak görebilirler. bu bölümü, öğrencilerin kelime dinleme ve okuma alıştırmaları yapmalarını sağlar.



Şekil 13: Öğrenme Kismi

### 3.4.2 Kelime Bulma

Bu ekranda hedef dili gözden geçirebilmek .resme bakıp ve eşleşen kelimeye tıklamak gerekir. Herhangi bir zamanda hata yapmış olursa ya cevap doğru ise aşağıda hem doğru hem hata simge olarak gösterilir.Oyun cevap geç olursa otomatik olarak sonraki sayfa geçiş yapacak ve yanlış hata olduğunu işaretlenecek oyunun bittiğinde kaç puan aldığını gösteren bir ekran çıkacak.Altakı şekil 14 örnek verilmiş.[12]



Şekil 14: Kelime Bulma

### 3.4.3 sürükle bırak

Çocuklar için sürükle bırak oyunu, çocukların düşünme, dokunsal ve ince motor becerilerini geliştirmelerine yardımcı olacaktır. hayvanları, renkler, sayılar ve meyveleri oynatarak öğrenirler. küçük çocuklar uzun süre oynarken eğlenebilirler. çalışma mantığı şu şekilde : sol taraftaki resimleri sağ tarafta karşılığı olan kelimelerle eşleştirmek ve her doğru eşleştirme 10 eklenecek aksi durumda toplam puandan 5 puan çıkarılacak .oyunun bittiğinde kaç puan aldığını gösteren bir ekran çıkacak.Aşağıdaki şekil 15 örnek verilmişti.



Şekil 15: sürekleme

#### 3.4.4 Yazma

Yazma, çocukların ustalaşması gereken önemli bir beceridir, ancak birçokları için zor olabilir. Yazım alıştırmaları yapmaya yardımcı olarak öğrencinin yazım deneyimini geliştiren birçok eğitim uygulamasıyla: eğlenceli, etkileşimli ve akılda kalıcı. Spelling Bee listesi, herhangi bir adayın araması gereken kapsamlı bir kelime listesidir. bu kelimeler rastgele olarak gelir ve üste çıkan boşluk yerlerele uygun kelime koymak .Aşağıdaki şekil 16 da gösterilmiş.



Şekil 16: Yazma

## 4 SONUÇLAR VE ÖNERİLER

Bu projede çocuklara yönelik bir ingilizce uygulaması yapıldı. Dart dili ve Flutter'ın sağladığı widget yapısının avantajları kullanıldı. Hazır paketler ile proje geliştirildi. Hitap edilen seviyede birkaç çocuklara uygulama gösterildi. Çocuklar uygulamayı beğendikleri, kullanmayı istedikleri sonucuna varıldı.

Bu projenin herkes kullanabilmesi için ilerleyen aşamalarında,dil kısmını daha fazla diller yüklenebilir ve daha fazla oyun entegre edilebilir.

## **5 EKLER**

<https://github.com/MahamatKabir/Flutter/tree/main/kalamna>



## KAYNAKLAR

- [1] Kelly, A. E. (2008). Reflections on the flutter Panel.
- [2] Gürsoy, K. (2010). flutter öğretiminde kullanılmasına ilişkin inanç ve tutumlarının incelenmesi. , Trabzon.
- [3] Nicolaidou, M., Philippou, G. (2003). Attitudes towards, Self-Efficacy and Achievement in Problem Solving. In M. A. Mariotti (Ed.), European Research in flutter III (pp.1-11). Italy: University of Pisa.
- [4] Ma, X., Kishor, N. (1997). Assessing the Relationship of flutter: A Meta-Analysis. Journal for Research in Education, 28, 26-47.
- [5] Philippou, G.N, Christou, C(1998). The Effects of a Preparatory flutter Program in Changing Prospective Teachers' Attitudes towards flutter. Educational Studies in computer, 35, 189-206.
- [6] Alakoç, Z. (2003). flutter teknolojik modern öğretim yaklasimleri. TOJET: The Turkish Online Journal of Educational Technology, 2(1).
- [7] Arslan, E. H. Bilgin, E. A. (2020). flutter Öğretiminde Teknoloji Kullanımı ve Video ile Öğretimin Teknoloji Tutumuna Etkisi . Dergisi, 3 (1) , 41-50 .
- [8] <http://fatihprojesi.meb.gov.tr/index.html#about> [Ziyaret Tarihi: 06/03/22]
- [9] <https://tr.wikipedia.org/wiki/Firebase> [Ziyaret Tarihi: 06/03/22]
- [10] <https://firebase.google.com/docs/auth/web/start> [Ziyaret Tarihi: 24/04/22]
- [11] [https://pub.dev/packages/rflutter\\_alert](https://pub.dev/packages/rflutter_alert) [Ziyaret Tarihi: 17/06/22]
- [12] [https://pub.dev/packages/animated\\_splash\\_screen](https://pub.dev/packages/animated_splash_screen) [Ziyaret Tarihi: 17/06/22]
- [13] <https://pub.dev/packages/lottie> [Ziyaret Tarihi: 17/06/22]

## **ÖZGEÇMİŞ**

### **KİŞİSEL BİLGİLER**

**Adı Soyadı** : Mahamat Kabir Souleyman Issein

**Uyruğu** : Çad

**Doğum Yeri ve Tarihi:** Çad 21/21/1998

**Adres** : Bilecik

**Telefon** : +905433044164

**e-mail** :

### **EĞİTİM DURUMU**

**Lisans Öğrenimi** : BŞEÜ Bilgisayar Mühendisliği Bölümü

**Bitirme Yılı** : 2023

**Lise** : Maarif tchadoturc

### **İŞ DENEYİMLERİ**

**Yıl** :

**Kurum** :

**Stajlar** :

### **İLGİ ALANLARI:**

**YABANCI DİLLER: türkçe,arabça,fransızca,ingilizce**

**BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER:**