

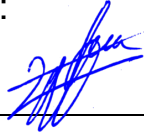
**15 / 08 / 2022 TARİHİNDEN 19 / 08 / 2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Tanışma Toplantısı Ve Staj Yapılacak Kurumun Belirlenmesi	1
SALI	Proje Platformu Ve Yazılacak Dilin Belirlenmesi	2
ÇARŞAMBA	Dart Programlama Diline Giriş	3
PERŞEMBE	Dart Programla Dilinde Koşul Yapıları , Operarölerinin Öğrenilmesi.	4
CUMA	Dart Dilinde Listeler Ve Döngülerin Öğrenilmesi ve Uygulama Yazılması	5

ÖĞRENCİNİN İMZASI

: 

KONTROL EDENİN İMZASI :

: 

**22/ 08 / 2022 TARİHİNDEN 26 / 08 / 2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Dart Dilinde Map Yapısı Ve Methodların Öğrenilmesi Ve Uygulama Yapılması	6
SALI	Dart Nesne yönelimli Programlamaya giriş ve Örnek Uygulama Yapılması	7
ÇARŞAMBA	Dart dilinde kalıtım ile Örnek Uygulama Yapılması	8
PERŞEMBE	Null safety hatası ve Fluttera Giriş	9
CUMA	Scaffold , Text, SizezBox	10

ÖĞRENCİNİN İMZASI

:

KONTROL EDENİN İMZASI

:

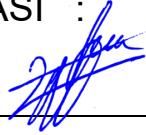
**29 / 08 / 2022 TARİHİNDEN 2 / 09 / 2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Button , ThemeData ,Color	11
SALI	30 ağustos zafer bayramı	
ÇARŞAMBA	Stattles Widget , Padding Ve Image Widgetlarını Kullanarak Flutter Uygulamasının Yapılması	12
PERŞEMBE	Listview Navigation ,Tabar Kullanrak Flutter Uygulaması Yapımı	13
CUMA	Flutter Ile Todo Uygulamasının Yapılması	14

ÖĞRENCİNİN İMZASI

: 

KONTROL EDENİN İMZASI :

: 

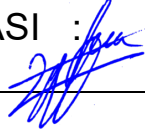
**05 / 09 / 2022 TARİHİNDEN 09 / 09 / 2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	To-Do uygulamasına devam	15
SALI	SQLite Araştırılması	16
ÇARŞAMBA	Veritabanı Ve Veritabanı Tablosu Oluşturma	17
PERŞEMBE	CRUD Metotları ekleme	18
CUMA	Todo Uygulamasının Geliştirilmesi	19

ÖĞRENCİNİN İMZASI

: 

KONTROL EDENİN İMZASI :

: 

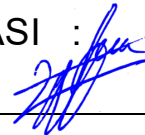
**12 / 09 / 2022 TARİHİNDEN 12 / 09 / 2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Veri Tabanı İle Uygulama Geliştirilmesi	20

ÖĞRENCİNİN İMZASI

: 

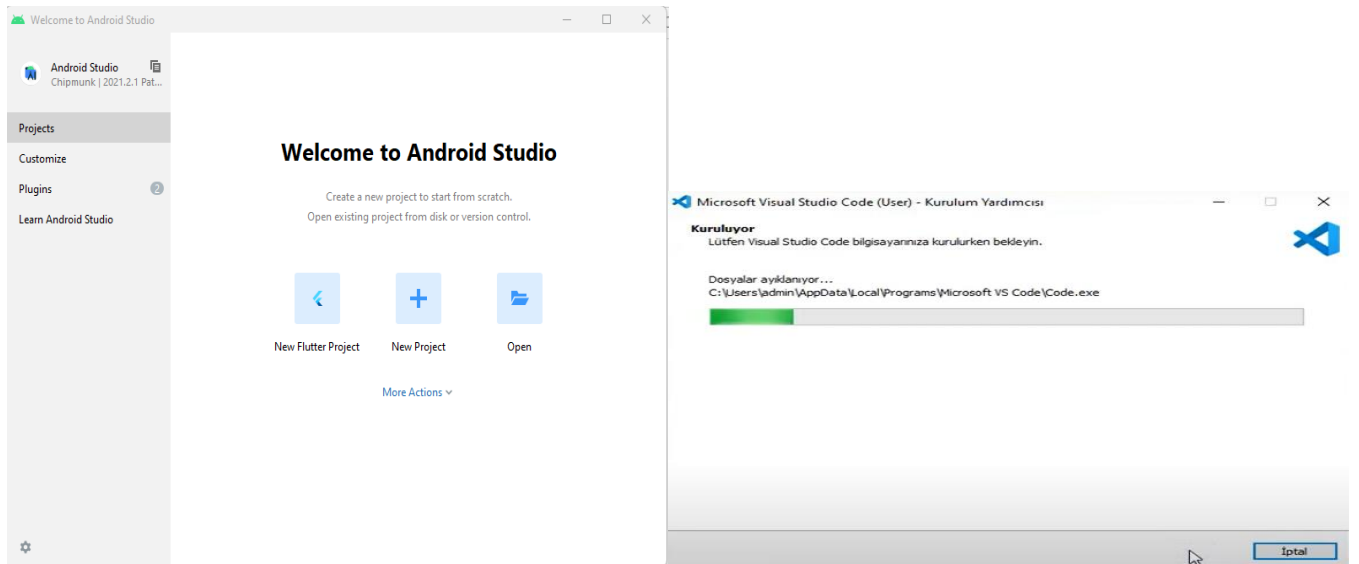
KONTROL EDENİN İMZASI :

: 

KISIM	Tanışma Toplantısı Ve Staj Yapılacak Kurumun Belirlenmesi	YAPRAK NO : 1
YAPILAN İŞ	Tanışma Toplantısı Ve Gerekli Uygulamaların Kurulumu	TARİH : 15-08-2022



KTÜ'deki ilk iş günümde sabah Kariyer merkezi insan kaynakları departmanında Derya hanım ile küçük bir toplantı gerçekleştirdik ve toplantı sonucunda KTÜ Bilgisayar Mühendisliği Bölümünde Öğr.Gör.Dr Zafer Yavuz beyin yanına gittim. Zafer beyde benim satjımın KTÜ uzaktan eğitim merkezinde yapmamın daha doğru olacağını söyledi. Ardında Uzaktan eğitim merkezine giderek mentörüm Fatih beyle tanıştım . Ardından mentörüm bana sekreter Fatma hanımın yanına gönderdi. Fatma Hanım Uzaktan eğitim merkezinin işleyişini anlattıktan sonra tekrar mentörümüm yanına gittim. Mentörümle küçük bir toplantı yaptıktan sonra mobil alanda proje geliştirmeye karar verdik . Mobil alanda hangi platform için proje yapacağıma ve hangi dille yazacağıma yarın karar vereceğini açıkladı.Mentörüm ardından beni proje labına gitmemi ve oradaki çalışacağım bilgisayara vscode ve android studio kurmamı istedi.



Android studioyu kurduktan sonra içine birtanede emulator kurulumu gerçekleştirdim. Bütün kurumları tamamlandıktan sonra gerekli sistem ayarlarını yaptım.

KISIM	Proje Platformu Ve Yazılacak Dilin Belirlenmesi	YAPRAK NO : 2
YAPILAN İŞ	Proje Platformu İçin Gerekli Kurumlar Yapildi.	TARİH : 16-08-2022



Sabah uzaktan eğitim merkezine geldiğimde mentörümün yanına gittim . Mentörüm Proje için kotlin, Xamarin, swift ve flutter konusunda araştırma yapmamı istedi. Öğlen arası olmadan yanına gitmemi ve yaptığım araştırmaları sunmamı istedi.Yapptığım araştırmalar sonucu öğle arası olmadan mentörümün yanına gittim . Mentörüm araştırmalarımı sunmamı istedi. Xamarinin türkçe kaynak sorunun ve geliştiricinin az olması swiftin sadece ios tabanlı uygulamalar yazabildiğini ve kotlinin futtera göre daha az esnek olduğunu belirttim.



Flutterın ise widget yapısının çok güçlü olduğunu swift ve kotlin gibi belirli kalıpları olmadığı için bize esnek sağladığını bunların yanında geliştiricisinin çok olduğunu ve google destekli olmasından bahsettim.

Tüm bunların yanında flutterın dart dili ile yazıldığı için öğrenimin kolay olabileceğini açıkladım. Dezavantaj olarak ise çok komponentin olduğunu ve bu komponentlerinde oldukça özelliklerinin olmasının kafa karışıklığına yol açabileceğini söyledim.Mentörümde benim yaptığım araştırmayı beğenerek flutter üzerinden mobil uygulama geliştirmemin daha doğru olacağını söyledi. Toplantımız bittikten sonra Bilgisayarımadaki vscode 'a dart ve flutter dil eklentilerini ekledim. Bu dillerin system ayarlarını yaptım..



KISIM	Dart Programlama Diline Giriş	YAPRAK NO : 3
YAPILAN İŞ	Dart programlama dili ile ilk uygulama yapımı.	TARİH : 17-08-2022

Sabah mentörüm ile kısa bir toplantı yaptık. Bana dart dili ile ilgili kurs ve kaynaklar verdi . Bunlara ek olarak bana dart için beş günlük bir yol haritası verdi. Toplantıdan sonra mentörümün vermiş olduğu kaynaklara baktım. Daha sonra dart dili ile uygun bir kurs buldum ve dart dilini öğrenmeye başladım.

```

1 void main() {
2   final String userName = "veli";
3
4   final int bankMoney = 100;
5   const String bankName = "VB Bank";
6
7   var userName2 = 'veli2';
8   var userName2Money = 15;
9
10  userName2Money -= 10;
11 // -----
12 // Bank name = "VB BANK"
13 // bank user 1 = "bank1musteri"
14 // bank user 1 in parasi 100.00
15 // bank1 musteriyeye kimse dokunmaz!
16 // yeni bir musteriy gelecek adi bank2musteri
17 // yeni bankaya gelenden bu bank user1 parasini cikartip ekrana gosterelim
18
19 const String bankNameSpecial = "VB Bank";
20 const String user1 = "Bank 1 musteriy";
21 const double user1Money = 100.00;
22
23 const String user2 = "Bank 2 msteriy";
24 int user2Money = 500;
25
26 user2Money = user2Money - user1Money.toInt();
27
28 print("User 2 money: $user2Money");
29 }

```

Gün sonunda bir dart projesi .dart uzantısı ile bittiğini başlangıç fonsiyonun void main() fonksiyonu olduğunu ekrana birşey basmak için print() fonksiyonun kullandığımızı , var değişkenin dynamic veri türü olduğunu const ve final keywordlerinin birbirlerine benzediğini ancak aralarındaki farkın flutterda uygulama ilk kez çalıştırıldığında ortaya çıktığını , int değişken tipi tam sayıları, double değişken türünün ondalıklı sayıları ve string değişken türünde metinsel ifadeleri tuttuğunu ,print fonksiyonu içinde metinsel olmayan bir tam sayı veya ondalıklı sayıyı \$ sembolü ile göstermeyi öğrendim.





KISIM	Dart Dilinde Listeler Ve Döngülerin Öğrenilmesi	YAPRAK NO : 5
YAPILAN İŞ	Dart Dilinde Listeler Ve Döngüler Kullanarak Uygulama Yapımı	TARİH : 19-08-1022

Stajımın beşinci gününde öğrenmem gereken konulara baktım. Ardından almış olduğum kursta öğrenmem gereken kısımları izledim. Ardından listeler ve fonskiyon yapılarıyla yazılmış örnek kodları inceledim.

```

1 void main() {
2 // Customers 100 30 40 60
3 // 35 tl den büyük olanları burada kredi verebiliriz yaz
4 // küçük olanlar kredi alamaz
5
6 List<int> moneyCustomerNews = [100, 30, 40, 60, -5];
7 moneyCustomerNews.sort();
8 for (int index = 0; index < moneyCustomerNews.length; index += 1) {
9   print('musteri parasi: ${moneyCustomerNews[index]}');
10  if (moneyCustomerNews[index] > 35) {
11    print('kredi hazır');
12  } else if (moneyCustomerNews[index] > 0) {
13    print('kredi veremeyiz ama bi bakalım');
14  } else {
15    print('malesef veremeyiz');
16  }
17 }
18 print('----');
19 for (int index = moneyCustomerNews.length - 1; index >= 0; index -= 1) {
20   print('musteri parasi: ${moneyCustomerNews[index]}');
21   if (moneyCustomerNews[index] > 35) {
22     print('kredi hazır');
23   } else if (moneyCustomerNews[index] > 0) {
24     print('kredi veremeyiz ama bi bakalım');
25   } else {
26     print('malasef veremeyiz');
27   }
28 }
29
30 }
31

```

Dart dilinde listeler aynı veri türünden birden çok veriyi tutmak için kullanılır.

Syntax yapısı List<veri tipi> listeadı = [liste elemanları] şeklindedir. Eğer veri tipi kısmına dynamic yazarsak bu sefer farklı veri tiplerindeki elemanları listede tutabiliriz.ancak bu kullanılabilecek bir durum değildir.Çünkü kodda verinin kontrolünü sağlamak zorlaşabilir.

Dart dilinde en çok kullanılan döngü olan for döngüsünün yapısı for(;;){ } şeklindedir.Buradaki ilk noktalı virgülden önceki kısma bir index değeri atanır,orta kısma koşul ve en son kısma ise indexin davranışı yazılır. Ardından bu koşulla bağlı olarak döngü bodysine girer ve buradaki kodları iterative şekilde koşar. İkinci döngü ise while döngüsüdür. While döngüsünde index dışarıda koşul while() parentezin içinde ve indexin davranışıda { } scope içinde yazılır.

KISIM	Dart Dilinde Map Yapısı Ve Methotların Öğrenilmesi	YAPRAK NO : 6
YAPILAN İŞ	Dart Dilinde Map Ve Methotlar Ile Uygulama Yapılması	TARİH : 22-08-2022

Stajımın altıncı gününde mentörüm vermiş olduğu plana sadık kalarak ilerlemeye devam ettim. Bugünün planında map yapısı ve methotlar vardı. Her zamanki gibi kursuma devam ettim ve bu yapılarla yazılmış örnek kodları inceledim. Ardından öğrendiğim bilgiler doğrultusunda uygulama yazdım.

```

1 void main() {
2     // müşteri geldi adı ahmet, parası 20
3     Map<String, int> users = {'ahmet': 20, 'mehmet': 30};
4     // müşteri ahmetin ne kadar parası var
5     print('ahmetin parası ${users['ahmet']}');
6     // müşteri kimler var senin elinde
7     for (var item in users.keys) {
8         print('${item} - ${users[item]}');
9     }
10    for (var i = 0; i < users.length; i++) {
11        print('${users.keys.elementAt(i)} - ${users.values.elementAt(i)}');
12    }
13    // benim bankayım. müşterimin birden fazla hesabı olabilir
14    // ahmet bey 3 hesabı var sırasıyla 100, 300, 200
15    // mehmet bey 2 hesabı var 30 50
16    // veli bey 1 hesap 30
17    // adamların hesaplarını kontrol et herhangi bir hesapta 150 tlden fazla olan varsa krediniz hazır de
18    print('-----');
19    final Map<String, List<int>> vbBank = {
20        'ahmet': [100, 300, 200],
21    };
22    vbBank['mehmet'] = [30, 50];
23    vbBank['veli'] = [30];
24    for (var item in vbBank.keys) {
25        // banknın tüm elemanları
26        for (var money in vbBank[item]) {
27            // kullanıcının hesaplarını dolaşıyorum
28            if (money > 150) {
29                print('kreditin hazır');
30                break;
31            }
32        }
33    }
34 }

```

Map yapısı key value prensibiyle çalışır. Syntax yapısı Map<veritürü1, veritürü2> mapismi=(veritürü1:veritürü2,...) şeklindedir. Birinci veritürü keydir. Valuelar ise birden çok olabilir.

```

// dart programı
void main() {
    // benim bir müşterinin var parası var mı yok mu kontrol eder misin
    final int userMoney = 0;
    controllUserMoney(userMoney, 0);
    // if (userMoney > 5) {
    //     print('parası var');
    // } else {
    //     print('parası yok');
    // }

    // benim bi müşterinin daha geldi onada bakar mısın
    final int user2Money = 5;
    controllUserMoney(user2Money, 5);
    // if (user2Money > 5) {
    //     print('parası var');
    // } else {
    //     print('parası yok');
    // }

    // ya dedi acaba 0 değilse on azından bi şti si olsun

    // yeni müşteri gelir parası 50 tl
    // parasını dolar yapmak istiyor ve kaç dolar yaptığını merak ediyor
    final newUserMoney = 50;
    print(newUserMoney / 13);

    int result = convertToDolar(newUserMoney);

    print(result);
    if (result > 0) {}

    // bana bir dolar hesaplama yap eğer ben sana söylemezsem yeni bir durum var diye her zaman 13 al

    final newResult = convertToStandartDolar(100, dolarIndex: 13);
    final newResult2 = convertToStandartDolar(100);
    final newResult3 = convertToEuro(userMoney: 500);
    sayHello('aa');
}

void controllUserMoney(int money, int minimumValue) {
    if (money > minimumValue) {
        print('para var');
    } else {
        print('para yok');
    }
}

int convertToDolar(int userMoney) {
    return userMoney ~/ 13;
}

int convertToStandartDolar(int userMoney, {int dolarIndex = 14}) {
    return userMoney ~/ dolarIndex;
}

int convertToEuro({required int userMoney, int dolarIndex = 14}) {
    return userMoney ~/ dolarIndex;
}

```

Methotların yazılmış bir fonksiyonu tekrar tekrar kullanabilmemizi sağlar. Parametrelili ve parametresiz olabilirler. Syntax Yapısı veritürü isim(){} eğer veri türü void ise methot bir değer döndürmez ancak veri türü void değilse return keywordü ile fonksiyonun tanımlanan veri türünü döndürür. Eğer parametreler Verilirken parameter{} scobu içerisinde verilmişse opsiyonelli demektir. Çağırıldığı yerde parametrelere değer atanırken; parametre isimleri kullanılır.

KONTROL SONUCU **7.9**

KISIM	Dart Nesne yönelimli Programlamaya giriş	YAPRAK NO : 7
YAPILAN İŞ	Dart oop prensibi kullanarak uygulama yapılması	TARİH : 23-08-22

Bugün mentörümün vermiş olduğu takvimin artık sonlarına doğru gelmiştim. Bugün claslar yapısını öğrenmem gerekiyordu. Önceki günlerdeki gibi kursumdaki ilgili videoları izledim ve class yapısı ile oluşturulmuş kodları inceledim.

```

1 // adi olmak zorunda
2 // parasi olmak zorunda
3 // yasini vermeyeibilir
4 // citiysini vermeyeibilir
5 // city yoksa ist say
6 // id degisine sadece bu sinif icinden eriseiblsin
7 class User {
8     // ozellikleri neler
9     late final String name;
10    late int money;
11    late final int? age;
12    late final String? city;
13
14    late final String userCode;
15
16    late final String _id;
17
18    User(String name, int money, {required String id, int? age, String? city}) {
19        this.name = name;
20        this.money = money;
21        this.age = age;
22        this.city = city;
23        _id = id;
24        userCode = (city ?? 'ist') + name;
25    }
26
27    bool isSpecialUser(String id) {
28        return _id == id;
29    }
30
31    // bool isEmptyId() {
32    //     return _id.isEmpty;
33    // }
34
35    bool get isEmptyId => _id.isEmpty;
36 }

```

Class yapısı yazdığımız kodu daha da dinamikleştirmemizi sağlayan bir kere yaz çok kere kulan prensibi ile çalışır. Class yapısı sayesinde bir nesneye ait özellikler ve methotları yazabiliyoruz. Eğer bu nesnenin özelliklerini nesne oluştururken default olarak vermek istersek bu seferde constructure kullanmamız gerekiyor. Classların syntax yapısı class class\_ismi{ atributlar ve methotlar} şeklindedir. Constructure class ismi ile aynı olan methottur. Eğer constructure overloading fonkcion yapmak istersek class\_ismi.yeni\_isim() sytanxını kullanmamız gerekecekti.

KISIM	Dart dilinde kalıtım	YAPRAK NO : 8
YAPILAN İŞ	Dart dilinde kalıtım ile uygulama yapılması	TARİH : 24-08-2022

Bugün artık dart dilinde öğrenmem gereken son konuya geldim . Bugün dart dilinde kalıtım yapısını öğrenmem gerekiyordu. Kursumu tamamlayıp her zaman yaptığım gibi yine kalıtım yapısı kullanılarak geliştirilmiş kodları inceledim.

```
void main() {  
  final userNormal = User('vb', 15);  
  final usersBank = BankUser('name', 155, 123);  
  final usrSpecial = SpecialUser('asdas', 12312, 123123, 30);  
  
  userNormal.sayMoneyWithCompanyName();  
  usersBank.sayMoneyWithCompanyName();  
  usrSpecial.sayMoneyWithCompanyName();  
  
  print(usrSpecial.calculateMoney());  
  print(usrSpecial.money);  
}  
  
// paralarını görebilelim istiyorum benim banka ismimle birlikte  
  
abstract class IUser {  
  final String name;  
  final int money;  
  
  IUser(this.name, this.money);  
  void sayMoneyWithCompanyName() {  
    print('Ahmet- $money paraniz vardır');  
  }  
}  
  
class User extends IUser {  
  final String name;  
  final int money;  
  
  User(this.name, this.money) : super(name, money);  
}  
  
class BankUser extends IUser {  
  final int bankingCode;  
  
  BankUser(String name, int money, this.bankingCode) : super(name, money);  
  
  void bankSpecialLogic() {  
    print(money);  
  }  
}
```

Dart dilinde miras işlemi yaparak sınıflarımızı dahada özelleştiriz. Miras alan sınıf en esnek sınıf olurken miras veren sınıf en az özelliklere sahip sınıf olur.

Syntax yapısı class class\_ismi extends miras\_alinan\_class\_ismi{ } şeklindedir.

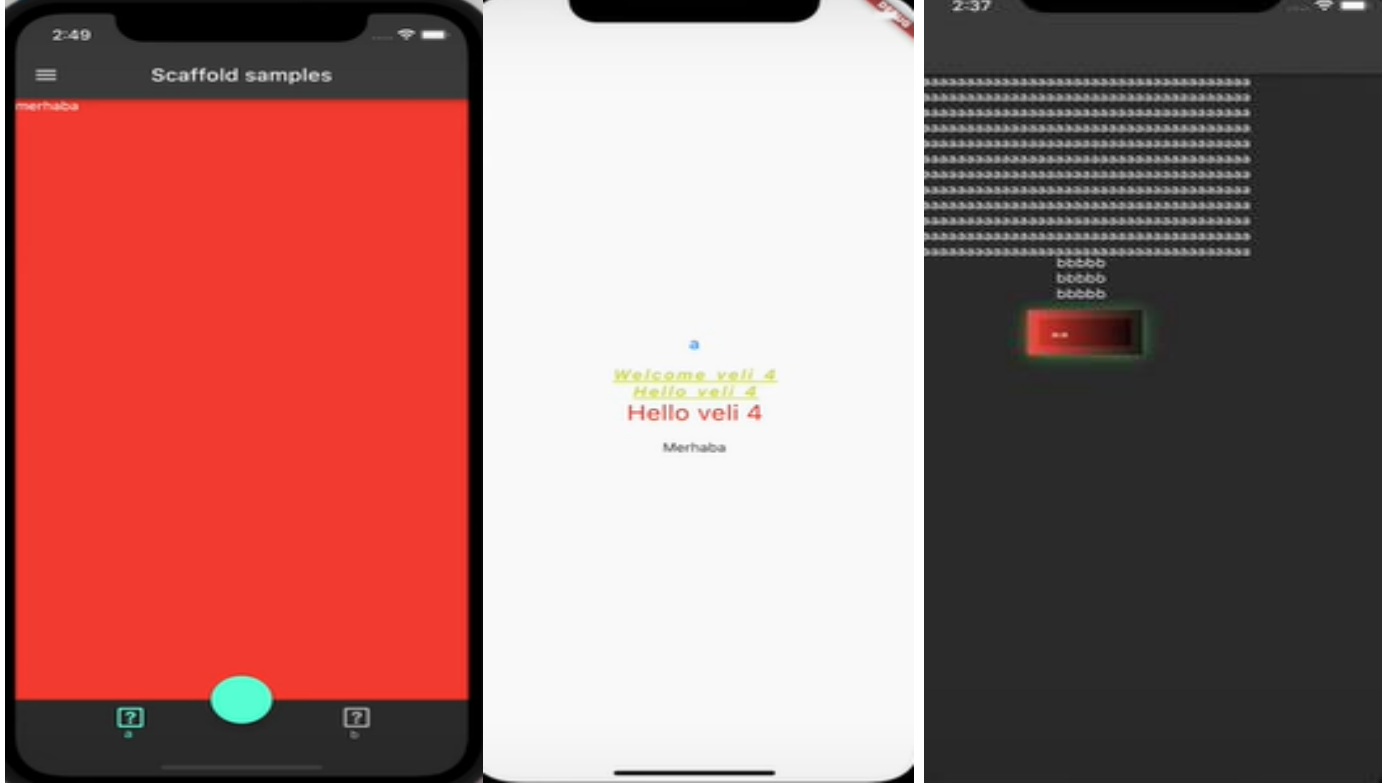
Abstract classlar soyut classlardır. İçerisinde attributlar ve methodlar bulundurabilir. Ancak kendisinden nesne üretilmez.

Dart dilinde her sınıf ancak tek bir sınıftan türetilir. Bu yüzden bir sınıf , birden fazla sınıftan miras alamamaktadır. İşte interface ile bu durumu aşabiliyoruz.

KISIM	Null safety hatası ve Flutter	YAPRAK NO : 9
YAPILAN İŞ	Null safety hatasının çözümü ve Flutterın Araştırılması	TARİH : 25-08-2022
<p>Bugün öğlen arasına kadar dart dilinde şimdiye kadar öğrendiklerimi tekrar ettim. Daha sonra nullsafety hatasını araştırdım . Nullsafety hatası bir değişkeni tanımlarken değer atamadığımız zaman ortaya çıkan bir hatadır. Ve bu hatayı önlemek için veritürünün sonu ? eklememiz gerekmektedir veya veri türünün önüne late keywordunu eklememiz gerekmektedir.</p> <div></div> <p>Bugün öğle arasından sonra mentörümün tekrar yanına gittim . Tamamladığım görevleri mentörümle paylaştım. Ardından mentörüm artık flutter kısmına geçmemi söyledi. Flutter ile ilgili bana kaynaklar verdi ve bu süreç içinde yedi günlük bir zaman tanıdı.</p> <p>Mentörümün yanından ayrıldıktan sonra flutter ile ilgili kurları araştırdım. Uygun gördüğüm bir kursa başladım. Kursun ilk kısımları flutter hakkında daha detaylı bilgiler vermekteydi.</p> <p>Flutter crossplatfor özelliği sayesinde hem android hem ios hemde web tabanlı uygulamalar yapılabilir. Bunlara ek olarak kotlin ve swift dillerinden farklı olarak bir toolsu uygulamanın istediğimiz yerine koyabilme özgürlüğünü bize sağlar.</p> <p>Son olarak bir widget yapısının olduğu ve bu widget yapısında ağaç şeklindedir.</p>		
KONTROL SONUCU Z.Y		

KISIM	Scaffold , Text, SizezBox	YAPRAK NO : 10
YAPILAN İŞ	Scaffold , Text, SizezBox kullanarak flutter uygulaması yapıldı	TARİH : 26-08-2022

Günüme öğrenilecekler listeme bakarak başladım. Ardından başlamış olduğum kurstan Scaffold, Text Sizezbox komponentlerine çalıştım. Kurs videolarına ek olarak api.flutter.dev sitesinden scaffold , text ve sizezbox komponentlerini inceledimi.



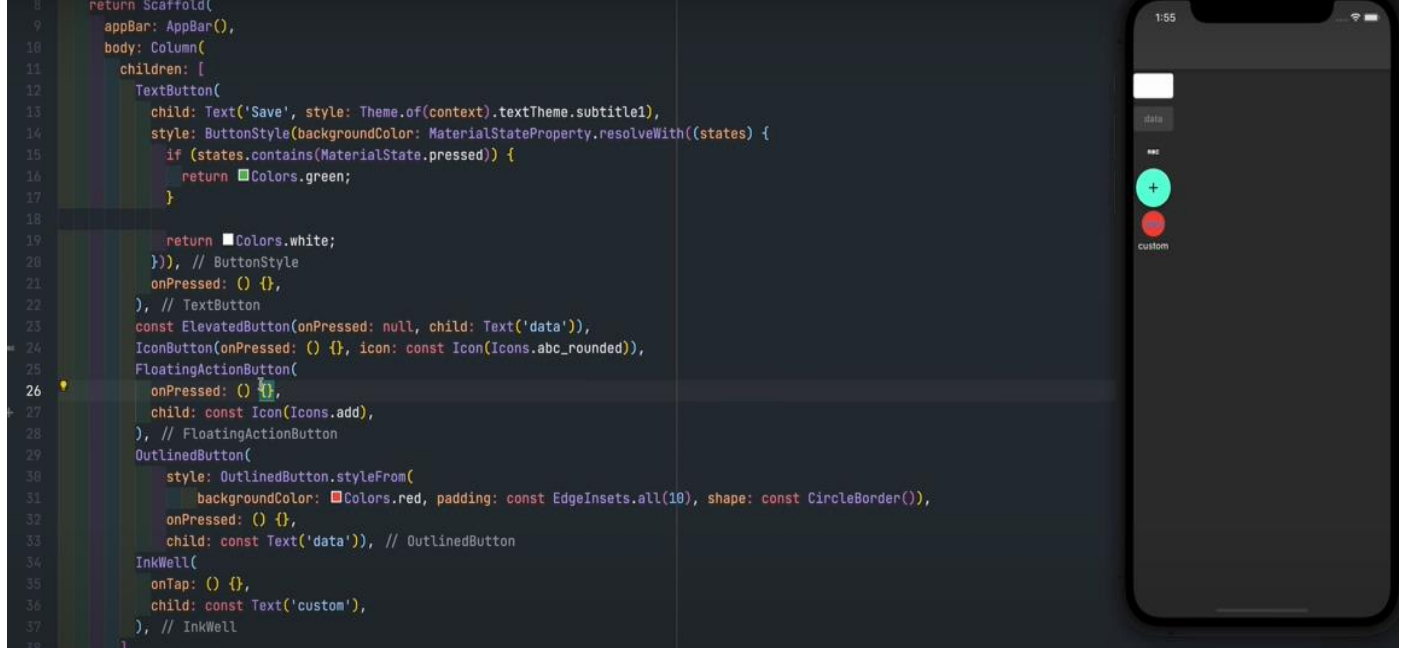
Scaffold appBar,body ve button shetten oluşur. AppBar uygulamayı açtığımız zaman uygulamanın üst kısmındaki bşlık kısmını , body appBar ile en allta kalan buton kısmı arasındaki bölgeyi buttomSheette butonların tasarımınlarını yaptığımız alanlardır.

Text , uygulamamız içerisindeki yazı tiplerinin özellikleriniayarladığımız sınıftır. Maxline,overflow,textAlignment,style ençok kullanılan özellikleridir. Style kısmında ise wordSpacing,decoration,fontstyle,color ençok kullanılır.

SizedBoxlar uygulamada scaffoldun body kısmında daha çok kullanılır. SizedBoxlar sayesinde body kısmında belirli alanlar oluşturup daha sonra buraya koyacağımız resimlerin veya yazacağımız textlerin taşmasını önleriz. Bu sayede uygulamanın tümünde bir bütünlük sağlarız.

KISIM	Button , ThemeData ,Color	YAPRAK NO : 11
YAPILAN İŞ	Button , Themedata ,Color Kullanarak Flutter Uygulaması Yapılması	TARİH : 29-08-2022

Bugün güne görev listeme bakarak başladım . Ardından Button ThemeData Color komponentlerine çalıştım. Kurs videolarına ek olarak api.flutter.dev sitesinden Button , ThemeData ve Color komponentlerini inceledim. Örnek olarak bir kaç proje inceledim.



TextButton gölgesiz basit bir butondur. onPressed() { } child propertylerinin kullanılması zorunludur.

ElevatedButton() dikdörtgen bir gövdeye sahiptir. textButton gibi onPressed ve child propertylerini kullanılması zorunludur.

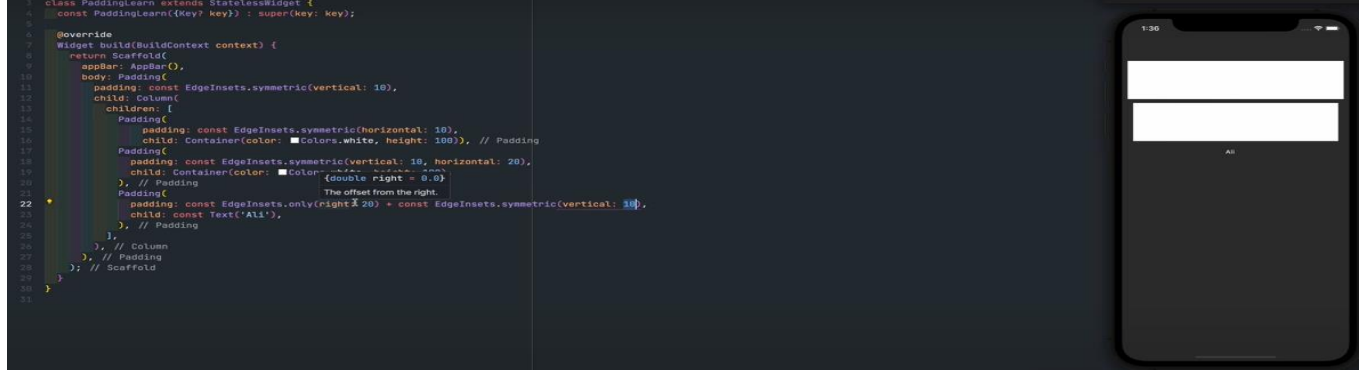
OutlinedButton() uygulamamızda dikkat çekmek istediğimiz butonları belirtmek için kullanırız

ThemeData uygulamanın genel temasını değiştirmek veya kullandığımız bir butona gölge 3 boyut katabilme işlevlerini yapar.

Color ise uygulamada kullandığımız tüm komponentler için renk belirlemede kullanırız.

KISIM	Stateless Widget , Padding Ve Image Widgetlar	YAPRAK NO : 12
YAPILAN İŞ	Stateless Widget , Padding Ve Image Widgetlarını Kullanarak Flutter Uygulamasının Yapılması	TARİH : 31-08-2022

Bugün mentörümle küçük bir toplantı yaparak güne başladım. Mentörüm çalışmalrıma aynen devam etmemi söyledi. Ardından görev listeme baktım ve Stateless widget padding image widgetlara çalışmaya başladım. Eğitim videolarını izledikten sonra örnek uygulamalar inceledim.



Stateless widget tüm widgetların miras aldığı sınıftır. Ağaç yapısının başındaki widgettır.

Parent widgetımızda bulunan child widgetıyla arasındaki uzaklığı padding widget'ıyla belirleriz. Burada EdgeInsets fonksiyonuyla yaparız.

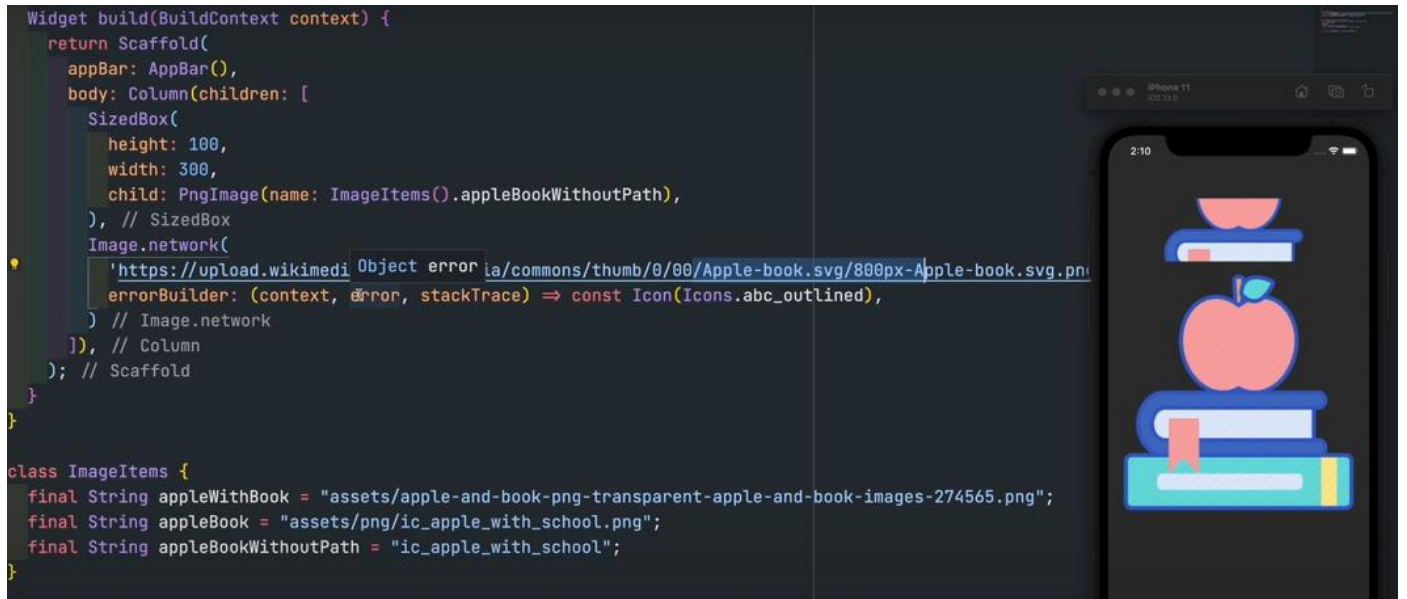
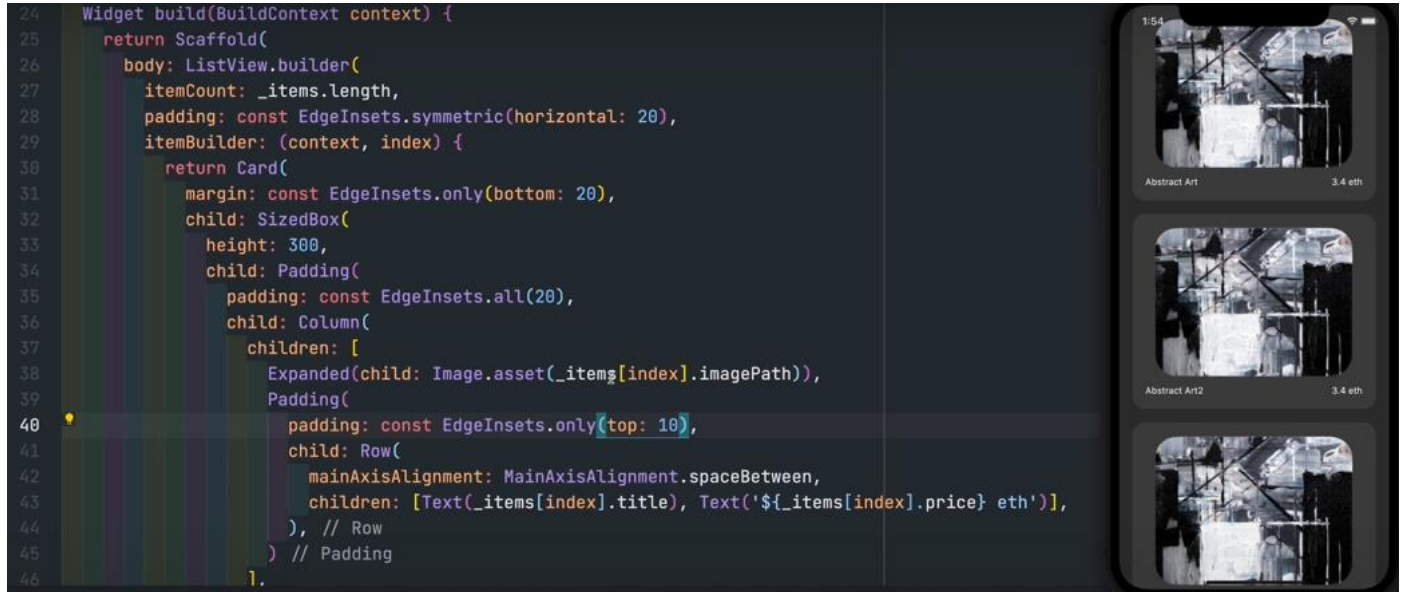


Image widget farklı kaynaklardan ve biçimlerden görselleri görüntülemek için kullanılır. Görseli uygulamada kullanmak için assets adında bir klasör oluştururuz ve kullanacağımız resimleri bu klasöre atarız ardından pubspec.yml dosyasına gidip yorum içinde bulunan assets dosyasını yorum satırından çıkarmamız gerekmektedir.



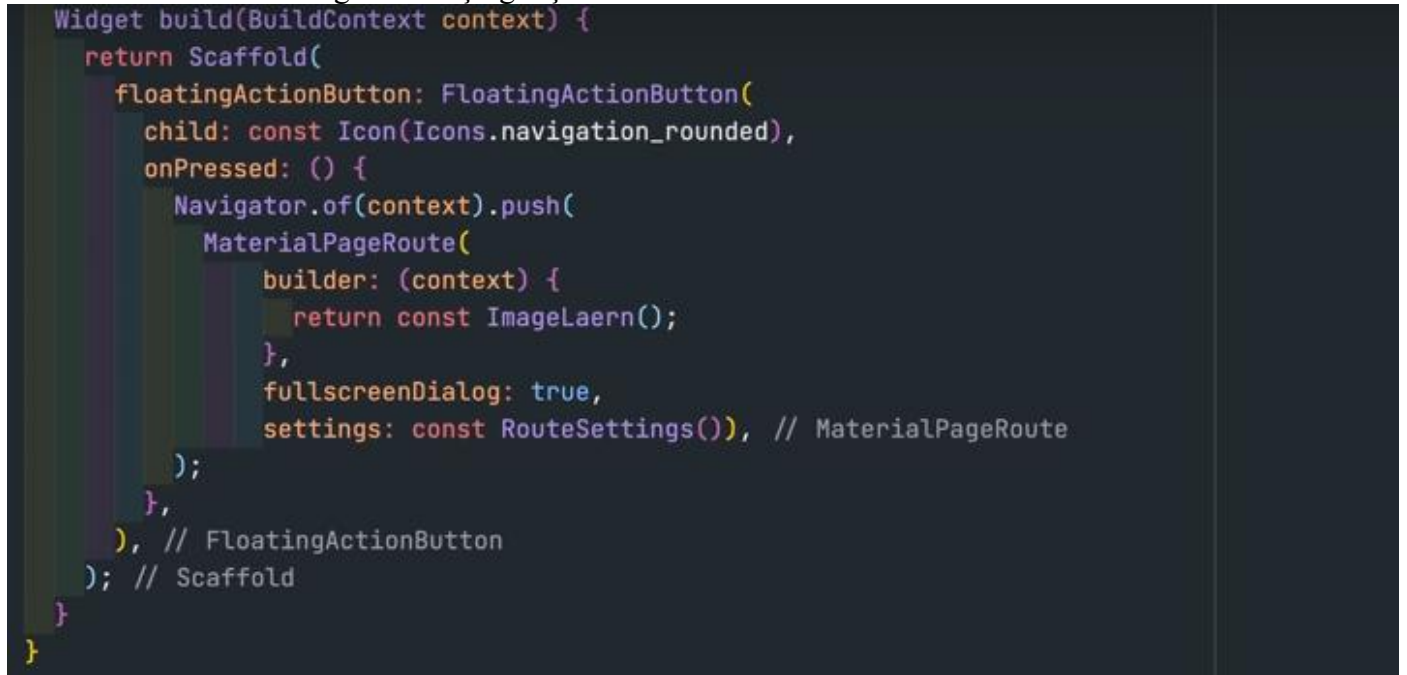
KISIM	List, Navigationa ,Tabbar	YAPRAK NO : 13
YAPILAN İŞ	Listview Navigation ,Tabar Kullanrak Flutter Uygulaması Yapımı	TARİH : 01-09-2022

Bugun flutter kursunun sonuna yaklaştım artık . Gorev listeme baktım eğitim videolarımın hepsini bitirdikten sonra List ,Nvigation ve tabbar ile yazılmış başlangıç uygulamara baktım.



Listviewları birden fazla elemanı olan listeler olarak düşünebiliriz. Bu listemiz sadece yazılardan oluşabiliği gibi karışık türden verilerde de oluşturulabilir.

Listview birden fazla widgettan oluştuğu için childen alır.



Navigasyon bir ekrandan başka bir ekrana geçmemizi sağlayan yönteme verilen adıdır. 3 çeşittir. MaterialPageRoute , Route Map ve Generated routesdir.

KISIM	Todo Uygulaması	YAPRAK NO : 14
YAPILAN İŞ	Flutter Ile Todo Uygulamasının Yapılması	TARİH : 02-09-2022

Güne mentörümün yanına giderek başladım. Flutter için vermiş olduğu kaynakları ve almış olduğum kursu tamamladığımı söyledim. Mentörüm bugün ve yarın için öğrendiklerimi tekrar amaçlı basit uygulama yapmam gerektiğini söyledi ve toplnatımızı bitirdik.

Projeyi seçmeden önce internette araştırma yaptım ve to uygulaması yapmaya karar verdim. Sonra daha önce yapılmış bir kaç to-do uygulaması tasarımını inceledim ve ardından uygulamayı yazmaya başladım.

Önce uygulama çerçevesini oluşturdum. Scaffold'un body parametresini aktifleştirip AnaEkran widget'ını koydum. Stateful widget ile AnaEkran widgetini oluşturdum ve Container'ın içine bir Column widget koyarak children parametresini aktif ettim Sonra Column'un içine flexible widget textfield 2\*raisedbuttondört widget koydum.

```

51 Widget build(BuildContext context) {
52   return Container(
53     child: Column(
54       children: <Widget>[
55         Flexible(child: null),
56         TextField(
57           controller: t1,
58         ), // TextField
59         RaisedButton(
60           onPressed: elemanEkle,
61           child: Text("Ekle"),
62         ), // RaisedButton
63         RaisedButton(
64           onPressed: elemanCikar,
65           child: Text("Çıkar"),
66         ), // RaisedButton
67       ], // <Widget>[]
68     ), // Column
69   ); // Container
70 }
71 }

```

Ardından ikinci bir class oluşturdum . Clasin içine bir tane TextEditingController nesnesi , bir tane boş bir liste iki tane de fonksiyon

Fonksiyon ekledim. Fonksiyonlar eleman ekleme ve elemna çıkarma işlevini görecekler. Eleman ekle fonksiyonun setState içerisine .add metotudu ile ekleme işlemi yapıyoruz. Eleman çıkar fonksiyonun setState içerisine ise .remove method ile çıkarma işlemi yapıyoruz. Projedeki boş listeyi alışveriş listesi olarak adlandırdım. Burada ki ekleme çıkarma işlemleri alışveriş listesini etkilemektedir.

KISIM	To-Do uygulaması	YAPRAK NO : 15
YAPILAN İŞ	To-Do uygulamasına devam	TARİH : 05-09-2022

Güne todo uygulamsında kaldığım yerden devam ettim. Flexieble Widgetı doldurmaya başladım. Flexible'in child parametresine bir ListView.builder verdim. ListView.builder alisverisListesi'ni kullanarak aşağı kaydırılabilen ve sadece o an gördüğümüz elemanları ekrana çizerek çok sayıda liste elemanını performanslı şekilde gösterebilen liste görünümü oluşturunca bir widget. Bunu verince otomatik olarak kod editörümüz itemBuilder parametresini aktif ediyor. İtem/madde oluşturunca bir parametre bu. alisverisListesi'ni kullandığımdan dolayı Flutter bize ekranda alt alta maddeler hâlinde bir liste yapısı gösterecek.

```

Widget build(BuildContext context) {
  return Container(
    child: Column(
      children: <Widget>[
        Flexible(
          child: ListView.builder(
            itemCount: alisverisListesi.length,
            itemBuilder: (context, indeksNumarasi) => ListTile(
              subtitle: Text("Alışveriş Listesi"),
              title: Text(alisverisListesi[indeksNumarasi]),
            ), // ListTile
          ), // ListView.builder
        ), // Flexible
        TextField(
          controller: t1,
        ), // TextField
        RaisedButton(
          onPressed: elemanEkle,
          child: Text("Ekle"),
        ), // RaisedButton
      ],
    ),
  );
}

```

Alışveriş Listesi Uygulaması

ekmek
Alışveriş Malzemeleri

su
Alışveriş Malzemeleri

çilek
Alışveriş Malzemeleri

ekmek
Alışveriş Malzemeleri

ekmek
Alışveriş Malzemeleri

ekmek
Alışveriş Malzemeleri

Ekle

Çıkar

itemBuilder parametresinin aldığı fonksiyon yapısının parantezleri içindeki argümanlardan biri olan tam sayı türündeki sıraNumarasi argümanı alisverisListemin index değerini tutuyor. Bu değer TextWidget içindeki yapıya iletiliyor ve alisverisListesinin elemanları ekranda alt alta diziliyorlar.

ListView.builder'in itemCount parametresini de aktif ettim. Bu parametre bize liste görünümünün uzunluğunu soruyor. Vermediğim çoğu durumda hata aldım. Bu yüzden buraya alisverisListesi'nin eleman sayısını .length metodu ile verdim.

KISIM	SQLite	YAPRAK NO : 16
YAPILAN İŞ	SQLite Araştırılması	TARİH : 06-09-2022




Bugün mentörümle kıs bir toplantı gerçekleştirdim. Yapmış olduğum uygulamayı mentörümüne sundum. Ardından mentörüm veritabanıyla çalışan bir uygulama yazmamı söyledi. Bunun içinde SQLite'i araştırmamı istedi.


SQLite verileri yerel olarak depolamanın en yaygın yöntemlerden biridir. SQLite'i bir Flutter projesinde kullanmak için projedeki pubspec.yaml dosyasındaki dependencies kısmının altına , sqflite ve path\_provider paketlerin güncel hallerini eklememiz gerekir.

```
dependencies :  
  
  sqflite: "güncel versiyon",  
  path_provider: "güncel versiyon"
```

Bu paketi yükledikten sonra veri tabanını oluşturmak için openDatabase() methotudunu kullanmamız gerekir

openDatabase() metodu; veritabanı dosyasının yoluna ulaşılmasını sağlayan, veritabanının taşınmasını ve yükseltilmesini kolaylaştıran, veritabanının sürümü ve veritabanı başarıyla oluşturulduğunda, çağrılacak bir geri çağırma işlevini gerçekleştirebilen bazı parametreleri sağlar.

KISIM	Sqlite Devam	YAPRAK NO : 17
YAPILAN İŞ	Veritabanı Ve Veritabanı Tablosu Oluşturma	TARİH : 07-09-2022
<p>Bugün veri tabanında tablo nasıl oluşturulacağını öğrenerek başladım. Veritabanında tablo oluşturmadan önce createDatabase() ile veritabanı oluşturmamız gerekir.</p> <p>Veri tabanı oluşturduğumuz zaman async keywordü ile birlikte kullanmamız gerekir. Async keywordü databasesin asenkron çalışmasını sağlar ve bu sayede database üzerinde birden fazla işlem eş zamanlı olarak yapılabilir. Aync keywordu kullanıldığı methotarin başına await keywordü de gelmesi gerekir.</p> <pre> createDatabase() async{   String databasesPath = await getDatabasesPath();   String dbPath = join(databasesPath, 'my.db');    var database = await openDatabases(dbPath, version: 1, onCreate: populateDb);   return database; } </pre> <p>Database oluşturulduktan sonra openDatabase içine verdiğimiz populateDb() fonksiyonunu tanımlamamız gerekir.</p> <pre> void populateDb(Database database, int version) async {   await database.execute("CREATE TABLE Customer ("     "id INTEGER PRIMARY KEY,"     "first_name TEXT,"     "last_name TEXT,"     "email TEXT"     ")"); } </pre> <p>populateDb() fonksiyonu tamamladıktan sora body kısmına database.execute() methotdunun içine “CREATE TABLE TABLO_ISMI( )” ve tablo üsminin içinide sütün isimlerini vermemiz gerekir.</p>		
KONTROL SONUCU 		

KISIM	Veritabanı	YAPRAK NO : 18
YAPILAN İŞ	CRUD Metotları ekleme	TARİH : 08-09-2022
<p>Bugün tüm veritabanlarında olan crud metotlarını araştırdım. CRUD Create,Read,Update ve Delete kelimelerinin baş harflerinden oluşur. Crate method ile yenibir veritabanı oluşturabilir veya var olan bir veri tabanına tablo ekleyebiliriz tablo eklerken insert metodunuda kullanmak zorundayız.</p> <pre> Future&lt;int&gt; createCustomer(Customer customer) async {     var result = await database.insert("Customer", customer.toMap());     return result; } </pre> <p>Read operasyonu veritabanındaki birinci parametreye göre verilerin okunmasını sağlar. Query() metodu ile metod tablonun ismini ve sütunlarını aldıktan sonra veriyi döndürür.</p> <pre> Future&lt;List&gt; getCustomers() async {     var result = await database.query("Customer", columns: ["id", "first_name", "last_name", "email"]);      return result.toList(); } </pre> <p>Uptade Operasyonu where ifadesinde belirtilen belirli bir elemanın güncellenmesini sağlar. query() metoduyla benzerdir. Bu metodu verilerin güncellenmesi için kullanırız.</p> <pre> Future&lt;int&gt; updateCustomer(Customer customer) async {     return await database.update("Customer", customer.toMap(), where: "id = ?", whereArgs: [customer.id]); } </pre> <p>Delete Operasyonu veritabanından belirtilen elemanın silinmesini sağlar.</p> <pre> Future&lt;int&gt; deleteCustomer(int id) async {     return await database.delete("Customer", where: 'id = ?', whereArgs: [id]); } </pre>		
KONTROL SONUCU 		



KISIM	Veri Tabanı İle Uygulama Yapılması	YAPRAK NO : 19
YAPILAN İŞ	Todo Uygulamasının Geliştirilmesi	TARİH : 09-09-2022

Bugün güne mentörümle toplantı yaparak başladım. Toplantıda veritabanı ile yapmış olduğum çalışmaları gösterdim. Ardından mentörüm daha önce yapmış olduğm todo uygulamasının bu sefer veri tabanı ile yapmamı istedi.

Uygulamamı 2 sayfa olacak şekilde tasarladım ; ilki anasayfamız olan Todo'ların listelendiği sayfa, ikincisi ise Todo'ların detayının gösterildiği ve üzerinde değişiklik yapılabilirdiği detay sayfası. Sayfanın sağ alt köşesinde yer FloatingActionButton ile yeni bir Todo ekleme işlemi yaptım. Listede yer alan ilgili Todo'nun üzerine tıklayınca da sayfanın detayına gidebiliyoruz. Detayda Todo'nun alanlarını göstereceğiz ve AppBar'ın üstünde yer alan PopupMenu ile

Güncelleme/Kaydetme/Listeleme/Geri Dönme işlemleri yapılabilecek. Ardından veritabanı paketlerini aktifleştirdim.

```

class DbHelper {
  static final DbHelper _dbHelper = DbHelper._internal();
  String tblTodo = "todo";
  String colId = "id";
  String colTitle = "title";
  String colDescription = "description";
  String colPriority = "priority";
  String colDate = "date";

  DbHelper._internal();

  factory DbHelper() {
    return _dbHelper;
  }

  static Database? _db;
  Future<Database?> get db async {
    if (_db == null) {
      _db = await initializeDb();
    }
    return _db;
  }

  Future<Database> initializeDb() async {
    Directory dir = await getApplicationDocumentsDirectory();
    String path = dir.path + "todos.db";
    var dbTodos = await openDatabase(path, version: 1, onCreate: _createDb);
    return dbTodos;
  }

  void _createDb(Database db, int newVersion) async {
    await db.execute(
      "CREATE TABLE $tblTodo($colId INTEGER PRIMARY KEY, $colTitle TEXT, " +
      "$colDescription TEXT, $colPriority INTEGER, $colDate TEXT " +
      ");");
  }

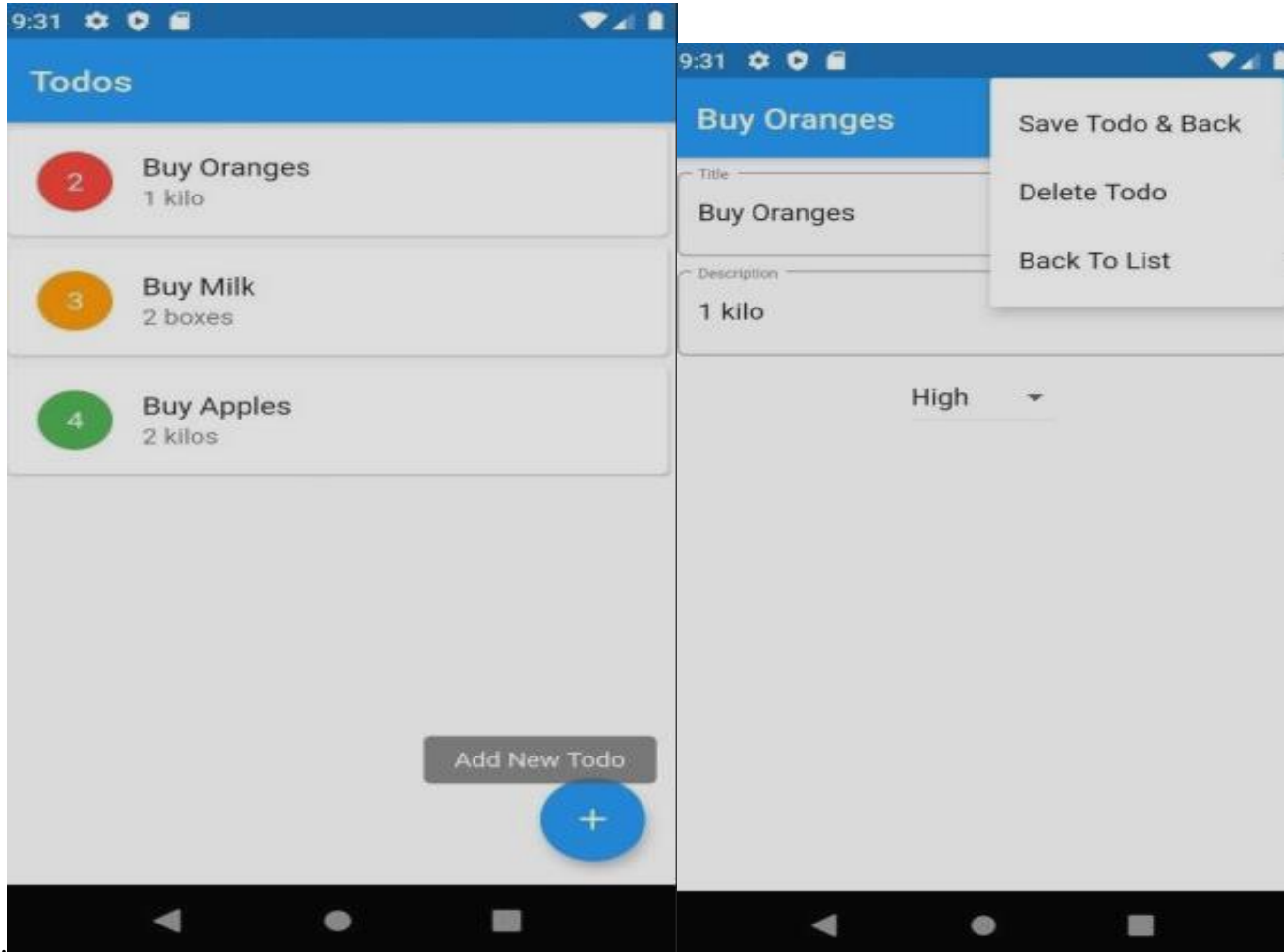
  Future<int> insertTodo(Todo todo) async {
    Database? db = await this.db;
    var result = await db!.insert(tblTodo, todo.toMap());
    return result;
  }
}

```

ToDo sınıfının 2 tane yapılandırıcısı(constructor) olacak.Database için dbelper.dart dosyasına DbHelper sınıfını oluşturdum. InitilazeDb metodu ile nesne oluşturdum ve ardından CretaDb ile veritabanımı oluşturdum. Todo adında da bir tane tablo oluşturdum.

KISIM	Veri Tabanı Ile Uygulama Geliştirilmesi	YAPRAK NO : 20
YAPILAN İŞ	Todo Uygulaması	TARİH : 12-09-2022

Bugün uygulamam kaldığım yerden devam ettim. Bir tane dummy Todo ekledim. Build metodu içerisine bir insertTodo ifadesi ekledim..InsertTodo bir çalıştıktan sonra silinecek ve veritabanına kaydolacak.Todoları Listelemek için ListView widgetını kullandım.todoListItems metodu içinde ListView widgetı oluşturdum. itemCount için verilen sayı kadar itemBuilder için geçilen metodu çağırarak. . itemBuilder parametresinin içinde todos listesinin ilgili position için verilen kaydı alınacak ve Text widget'ları içinde gösterilmesi sağlanacak. Listenin ilgili kaydını tıklama olayını ise onTap parametresi ile yönettim.



Uygulamayı bitirdikten sonra mentörümle tekrar görüştim. Son günüm olması sebebiyle staj boyunca yapmış olduğum çalışmalara tekrardan baktı ve beni tebrik etti. Ardından Staj sorumlum Zafer beyin yanına gittim yapmış olduğum çalışmaları tekrardan Zafer beye de gösterdim. Zafer beyde Fatih beyle sık sık görüştüğünü çalışmalarımı Fatih bey aracılığı ile takip ettiğini söyledi ve beni tebrik etti. Ardından staj evraklarımı imzalatması için benden staj defterimi ve staj sicil fişlerimi getirmemi istedi. Ardından bu belgeleri hazırlayıp getirdikten sonra evraklarımı imzalattım ve stajımı bitirdim.