

MOBİL PROGRAMLAMA ÖDEV6

22060388-ÖMER FARUK DERİN

GITHUB (odev6 klasörü) :

<https://github.com/omerfderin/22060388-OMER-FARUK-DERIN>

PROGRAM GÖRÜNTÜSÜ:

FORM

DEBUG

İsim

İsim giriniz

E-posta

E-posta giriniz

Şifre

Şifre giriniz

Gönder

FORM

DEBUG

İsim

İsim giriniz

Lütfen isminizi giriniz

E-posta

E-posta giriniz

Lütfen e-posta adresinizi giriniz

Şifre

Şifre giriniz

Lütfen bir şifre giriniz

Gönder

FORM

İsim

Ömer Faruk Derin

E-posta

faruk.derin@bil.omu.edu.tr

Şifre

.....

Gönder

Form başarıyla tamamlandı

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp( //Materialapp tanımlandı ve içerisine scaffold yerleştirildi
      home: Scaffold( //scaffold arkaplan rengi ayarlandı ve appBar eklendi
        backgroundColor: Colors.white, //appbar rengi ayarlandı ve başlık eklendi
        appBar: AppBar(
          backgroundColor: Colors.white,
          centerTitle: true,
          title: Text(
            'FORM',
          ),
        ),
      ),
      body: Center( //scaffold body kısmına center widgetı yerleştirildi ve içerisine container eklendi
```

```

        child: Container( //yatay ve dikey eksenlerde 25 piksel boşluk bırakıldı ve container boyutu ayarlandı
          padding: EdgeInsets.all(25),
          width: 400,
          height: 600,
          decoration: BoxDecoration(
            boxShadow: [ //Containera gölge eklendi
              BoxShadow(
                color: Colors.black.withOpacity(0.2),
                spreadRadius: 3,
                blurRadius: 5,
                offset: Offset(0, 3),
              ),
            ],
            borderRadius: BorderRadius.circular(10), //Container köşe geometrisi ayarlandı
            color: Colors.white, //container rengi ayarlandı ve kenar rengi eklendi
            border: Border.all(
              color: Colors.grey,
            ),
          ),
          child: RegistrationForm(), //Container içerisine yerleştirilecek sınıf tanımlandı
        ),
      ),
    );
  }
}

class RegistrationForm extends StatefulWidget {
  @override
  _RegistrationFormState createState() => _RegistrationFormState();
}

class _RegistrationFormState extends State<RegistrationForm> {
  final _formKey = GlobalKey<FormState>(); //Form içerisindeki özelliklerin kontrolü için globalkey tanımlandı
  final TextEditingController _nameController = TextEditingController(); //İsim kutucuğu içeriği kontrolcüsü tanımlandı
  final TextEditingController _emailController = TextEditingController(); //email kutucuğu içeriği kontrolcüsü tanımlandı
  final TextEditingController _passwordController = TextEditingController(); //şifre kutucuğu içeriği kontrolcüsü tanımlandı

  @override
  Widget build(BuildContext context) {
    return Form( //Yukarıda tanımlanan container içerisine form yerleştirildi ve
      key: _formKey, //anahtarına form kontrolü için key yerleştirildi
      child: Column( //Form içerisine column yerleştirildi ve elemanları arasında eşit boşluk olması sağlandı
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          Align( //column içerisine align widgetı tanımlandı ve merkeze konumlandırıldı
            alignment: Alignment.center, //içerisine bir text widgetı tanımlandı, font boyutu ve kalınlığı
            child: Text( //ayarlandı
              'İsim',
              style: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
          TextFormField( //İsim tutacak form alanı column içine kontrolcüsüyle beraber yerleştirildi
            controller: _nameController,
            textAlign: TextAlign.center, //Form girişinin yeri ayarlandı
            decoration: InputDecoration(
              hintText: 'İsim giriniz', //Form giriş kutucuğunun içerisine mesaj yerleştirildi
            ),
            validator: (value) { //İsim kutucuğunun içerişinin boş olup olmadığı kontrol edildi
              if (value == null || value.isEmpty) {
                return 'Lütfen isminizi giriniz';
              }
              return null;
            },
          ),
          Align( //Bu kısım isim kutucuğu ile aynı
            alignment: Alignment.center,
            child: Text(
              'E-posta',
              style: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
          TextFormField( //email form alanı ve kontrolcüsü tanımlandı isim kutucuğu ile aynı
            controller: _emailController,
            textAlign: TextAlign.center,
            decoration: InputDecoration(

```

```

        hintText: 'E-posta giriniz',
    ),
    keyboardType: TextInputType.emailAddress,
    validator: (value) {
        if (value == null || value.isEmpty) { //email kutucuğu boş kontrolü ve email format doğruluğu
            return 'Lütfen e-posta adresinizi giriniz';
        }
        final emailRegExp = RegExp(r'^[w-\.] + @ ( [w- ] + \. ) + [w- ] {2,4} $ ');
        if (!emailRegExp.hasMatch(value)) {
            return 'Geçerli bir e-posta adresi giriniz';
        }
        return null;
    },
),
Align( //şifre texti tanımlandı isim ve email kısmıyla aynı
    alignment: Alignment.center,
    child: Text(
        'Şifre',
        style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.bold,
        ),
    ),
),
TextFormField( //şifre form alanı ve kontrolcüsü tanımlandı isim ve email kısmıyla aynı
    controller: _passwordController,
    textAlign: TextAlign.center,
    decoration: InputDecoration(
        hintText: 'Şifre giriniz',
    ),
    obscureText: true, //şifrenin direkt olarak görünmesi yerine gizlenmesi sağlandı
    validator: (value) {
        if (value == null || value.isEmpty) { //Kutucuk boş mu kontrol edildi
            return 'Lütfen bir şifre giriniz';
        }
        if (value.length < 5) { //şifrenin en az 6 karakter olması sağlandı
            return 'Şifre en az 6 karakter olmalıdır';
        }
        return null;
    },
),
ElevatedButton(
    onPressed: () { //Form değerleri girildikten sonra doğruluk kontrolü yapıldı ve değerler doğru
        if (_formKey.currentState!.validate()) { //format ve değerlerdeyse kullanıcıya bir snackbar ile mesaj gösterildi
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(content: Text('Form başarıyla tamamlandı')),
            );
        }
    },
    child: Text('Gönder'), //Buton texti ayarlandı
),
],
),
);
}
}

```