

# Mobil Ders Seçme Uygulaması

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans Tezi

Yiğit Bal ORCID 0000-0000-0000-0000

Tez Danışmanı: Doç. Dr. Aytuğ Onan

Haziran 2023

# Yazarlık Beyanı

Ben, Yiğit Bal, Mobil Ders Seçme Uygulaması olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

* Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
* Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
* Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
* Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
* Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
* Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 24.06.2023

Mobil Ders Seçme Uygulaması

Öz

Günümüzde akıllı telefon kullanan insanların sayısı her geçen gün artmaktadır. Akıllı telefonların hayatımıza girmesiyle birlikte, mobil uygulamalar da hızla gelişmekte ve biz kullanıcıların hayatlarını kolaylaştırmayı amaçlamaktadır. Tüm farklı alanlarda mobil uygulamalar kullanılıyor, bu sayede de öğrencilerin ve okulların işlerini kolaylaştırabilecek olan mobil uygulamalar gün geçtikçe teknolojinin ilerlemesi ile hayatımızı kolaylaştırmaya devam etmektedir.

Bu bitirme tezimde; İzmir Kâtip Çelebi Üniversitesi öğrencilerinin, ders seçiminde kullanabilecekleri bir mobil uygulama yapımı amaçlanmıştır. Bu uygulamadaki temel amaç, uygulamada yayınlanan derslerin belirli bir kontenjana sahip olması ve öğrencilerin uygulamadaki genel başarılı notları girmeleri istenerek, belirtilen derslerin kontenjanlarının, başarılı sıralaması yapılarak önceliklendirilmesi temel amaçtır. Bu sayede de öğrencilerin ders kayıtları esnasında, başarı sıralamasına göre ders seçmelerine yardımcı olunmuştur.

**Anahtar Sözcükler:** Mühendislik, mobil, uygulama, ders, seçme, öğrenci.

Teşekkür

Yüksek lisans çalışma sürecimde değerli katkılarını ve desteklerini esirgemeyen, değerli hocam ve sayın danışmanım Sayın Doç. Dr. Aytuğ ONAN’a teşekkürlerimi ve şükranlarımı sunarım.

İçindekiler

Yazarlık Beyanı……………………………………………………………………2

Öz………………………………………………………………………………......3

Teşekkür……………………………………………………………………………4

1 Giriş………………………………………………………………………………6

2 Tezin İçeriği………………………………………………………………………7

3 Sonuç……………………………………………………………………………..29

4 Kaynaklar ………………………………………………………………………...30

5 Özgeçmiş ………………………………………………………………………....31

Giriş

Bitirme projemde ilk olarak Flutter ve Dart dilinden bahsetmek istiyorum; Flutter platformu hem Ios, hem Android hem de çeşitli Web ve Masaüstü uygulamaları geliştirmemize yarayan bir mobil uygulama hazırlamamıza yarayan geliştirme platformudur. Flutter’ın kullandığı dil olan bizim uygulamamız da Dart diliyle yazılmıştır. Öğrencilerin kullanabilmesi için her iki dilde de destek veren bir cross platform uygulaması olduğundan dolayı Flutter tercih edilmiştir.

Tezin İçeriği

Bitirme Tezimde Flutter’dan yararlanılarak Dart dilinde öğrencilerin ders seçmesi için bir uygulama geliştirildi. Uygulama şu anda ders seçme olarak aktif durumda olsa da geliştirilmeye devam edilecektir. Uygulama içinde şu anda 5 farklı sayfa bulunmaktadır. Bu sayfalardan ilk olarak login sayfası bulunmaktadır. Kullanıcılar yani öğrenciler, mobil uygulamalarını indirildikten sonra, İsim, Soyisim, Email ve Not ortalamalarını girerek ilk ekrandan giriş yapabileceklerdir.  
  
 ekran görüntüsü, metin, mobil telefon, mobil cihaz içeren bir resim

Açıklama otomatik olarak oluşturuldu

Daha sonra ise Ders seçmeleri için bir kural olan 150 AKTS kuralını göreceklerdir. Burada eğer öğrenciler (kullanıcılar) 150 AKTS’yi tamamlamadılar ise mobil uygulama ekranlarında bir uyarı ile karşılacaklar ve devam edemeyeceklerdir. Diğer bir yandan eğer 150 AKTS’yi tamamladılar ise, uygulamanın diğer ekranlarından devam edeceklerdir. İlgili ekranlar aşağıdadır.

ekran görüntüsü, mobil telefon, mobil cihaz, küçük alet içeren bir resim

Açıklama otomatik olarak oluşturuldu

Burada öğrenciler (kullanıcılar) eğer 150 AKTS’yi tamamlamadılar ise, Hayır butonuna tıkladıklarında ise aşağıdaki hata ile karşılaşacaklar ve uygulamaya devam edemeyeceklerdir.

metin, ekran görüntüsü, mobil telefon, küçük alet içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ders Seçimlerinin olduğu ekranda ise, belirli dönemde yer alan dersleri öğrenciler ekranlarında görebileceklerdir. İlgili ekran aşağıdadır.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Öğrenciler ( kullanıcılar ) , bu ekrandan ders seçimlerini yaptıklarında, diğer bir ekran olan bölümlerde yer alan dersleri ve bu derslere ait akademisyen bilgisi ( İsim, Soyisim ve Kontenjan bilgisi ) , ders açıklaması ve derse ait şube bilgilerini göreceklerdir. Bazı bölümlerde farklı seçeneklerde dersler yer almaktadır. Öğrenciler bu bölümdeki derslerden en fazla 1 şubeyi seçmeleri sağlanacak ve genel not ortalamaları baz kriter alınarak ilgili derse ait kontenjan ile derslerini seçebileceklerdir. Ekran görüntüsü aşağıdaki gibidir.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlgili ekranda öğrencilerin seçmesi için 1.Seçenek ve 2.Seçenek olarak iki farklı öğretim üyesi paylaşılmıştır. Şube bilgisi, Derse ait olan Öğretim Üyesi bilgisi, Açıklama ve son olarak kontenjan bilgileri iletilmiştir. Sadece öğrenciler Şube ile başlayan seçeneği seçebilirler. Seçim işleminden sonra ise son ekran olan final ekranı aşağıdaki şekilde paylaşılmıştır.

ekran görüntüsü, mobil telefon, metin, küçük alet içeren bir resim

Açıklama otomatik olarak oluşturuldu

Mobil uygulamaya ait Dart dilindeki tüm kodlar aşağıdaki gibidir;

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(LoginScreen());  
}  
  
class Student {  
 String firstName;  
 String lastName;  
 String email;  
 double gpa;  
  
 Student(this.firstName, this.lastName, this.email, this.gpa);  
}  
  
class LoginScreen extends StatelessWidget {  
 List<Student> students = [  
 //  
 ];  
  
 List<Student> sortedStudents = [];  
 List<double> sortedGPAs = [];  
  
 LoginScreen() {  
 sortedStudents = List.from(students)  
 ..sort((a, b) => b.gpa.compareTo(a.gpa));  
  
 sortedGPAs = List.from(sortedStudents.map((student) => student.gpa))  
 ..sort((a, b) => b.compareTo(a));  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Öğrenci Girişi'),  
 ),  
 body: Container(  
 padding: EdgeInsets.all(16.0),  
 child: LoginForm(sortedStudents, sortedGPAs),  
 ),  
 ),  
 );  
 }  
}  
  
class LoginForm extends StatefulWidget {  
 final List<Student> students;  
 final List<double> studentGPAs;  
  
 LoginForm(this.students, this.studentGPAs);  
  
 @override  
 \_LoginFormState createState() => \_LoginFormState();  
}  
  
class \_LoginFormState extends State<LoginForm> {  
 TextEditingController firstNameController = TextEditingController();  
 TextEditingController lastNameController = TextEditingController();  
 TextEditingController emailController = TextEditingController();  
 TextEditingController gpaController = TextEditingController();  
  
 void login() {  
 //  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => LessonsPage(),  
 ),  
 );  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Column(  
 children: [  
 TextField(  
 controller: firstNameController,  
 decoration: InputDecoration(  
 labelText: 'İsim',  
 ),  
 ),  
 TextField(  
 controller: lastNameController,  
 decoration: InputDecoration(  
 labelText: 'Soyisim',  
 ),  
 ),  
 TextField(  
 controller: emailController,  
 decoration: InputDecoration(  
 labelText: 'E-posta',  
 ),  
 ),  
 TextField(  
 controller: gpaController,  
 decoration: InputDecoration(  
 labelText: 'GPA',  
 ),  
 ),  
 SizedBox(height: 16.0),  
 for (int i = 0; i < widget.students.length; i++)  
 Text('${widget.students[i].firstName} ${widget.students[i].lastName}: ${widget.studentGPAs[i]}'),  
  
 SizedBox(height: 16.0),  
 ElevatedButton(  
 onPressed: login,  
 child: Text('Giriş Yap'),  
 ),  
 ],  
 );  
 }  
}  
  
  
  
  
  
  
class LessonsPage extends StatelessWidget {  
 List<String> lessons = [  
 'Biyomedikal Mühendisliği / Biomedical Engineering',  
 'Elektrik-Elektronik Mühendisliği / Electrical and Electronics Engineering',  
 'Harita Mühendisliği / Geomatics Engineering',  
 'İnşaat Mühendisliği / Civil Engineering',  
 'Makine Mühendisliği / Mechanical Engineering',  
 'Mekatronik Mühendisliği / Mechatronics Engineering',  
 'Metalurji ve Malzeme Mühendisliği / Metalurgical and Material Engineering',  
 'Petrol ve Doğalgaz Mühendisliği / Petroleum and Natural Gas Engineering',  
 //  
 ];  
  
 void handleEvetButton(BuildContext context) {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => LessonsSelectionPage(),  
 ),  
 );  
 }  
  
 void handleHayirButton(BuildContext context) {  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 return AlertDialog(  
 title: Text('Ders Seçimi için 150 AKTS\'yi Tamamlamanız Gerekmektedir !'),  
 actions: [  
 TextButton(  
 onPressed: () {  
 Navigator.*pop*(context);  
 },  
 child: Text('Tamam'),  
 ),  
 ],  
 );  
 },  
 );  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Ders Seçimi'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('150 AKTS\'yi Tamamladınız mı?'),  
 ),  
 ElevatedButton(  
 onPressed: () => handleEvetButton(context),  
 child: Text('Evet'),  
 ),  
 ElevatedButton(  
 onPressed: () => handleHayirButton(context),  
 child: Text('Hayır'),  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class LessonsSelectionPage extends StatelessWidget {  
 List<String> lessons = [  
 'Biyomedikal Mühendisliği / Biomedical Engineering',  
 'Elektrik-Elektronik Mühendisliği / Electrical and Electronics Engineering',  
 'Harita Mühendisliği / Geomatics Engineering',  
 'İnşaat Mühendisliği / Civil Engineering',  
 'Makine Mühendisliği / Mechanical Engineering',  
 'Mekatronik Mühendisliği / Mechatronics Engineering',  
 'Metalurji ve Malzeme Mühendisliği / Metalurgical and Material Engineering',  
 'Petrol ve Doğalgaz Mühendisliği / Petroleum and Natural Gas Engineering',  
 // Add other lessons here.  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Ders Seçimi'),  
 ),  
 body: ListView.builder(  
 itemCount: lessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(lessons[index]),  
 onTap: () {  
 if (lessons[index] == 'Biyomedikal Mühendisliği / Biomedical Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => BiomedicalEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Elektrik-Elektronik Mühendisliği / Electrical and Electronics Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => ElectricalEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Harita Mühendisliği / Geomatics Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => GeomaticsEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'İnşaat Mühendisliği / Civil Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => CivilEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Makine Mühendisliği / Mechanical Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => MechanicalEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Mekatronik Mühendisliği / Mechatronics Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => MechatronicsEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Metalurji ve Malzeme Mühendisliği / Metalurgical and Material Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => MetalurgicalEngineeringPage(),  
 ),  
 );  
 } else if (lessons[index] == 'Petrol ve Doğalgaz Mühendisliği / Petroleum and Natural Gas Engineering') {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => PetroleumEngineeringPage(),  
 ),  
 );  
 } else {  
 //  
 //  
 }  
 },  
 );  
 },  
 ),  
 );  
 }  
}  
  
class BiomedicalEngineeringPage extends StatelessWidget {  
 List<String> ahmetHocaLessons = [  
 'Şube1: Doç. Dr. Yalçın İşler (40/40)',  
 'Açıklama:',  
 'Grup#1: Doç. Dr. Yalçın İŞLER, Fiziksel İnsan Robot Etkileşimi İçin İstek Tanımlama ve Robot Manipülatörler Üzerinde Uygulanması',  
 ];  
  
 List<String> mehmetHocaLessons = [  
 'Şube7: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#7: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Biyomedikal Mühendisliği / Biomedical Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: ahmetHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(ahmetHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: mehmetHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(mehmetHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class ElectricalEngineeringPage extends StatelessWidget {  
 List<String> huseyinHocaLessons = [  
 'Şube8: Dr. Öğr. Üyesi Hüseyin YEŞİLYURT (40/40)',  
 'Açıklama:',  
 'Grup#8: Dr. Öğr. Üyesi Hüseyin YEŞİLYURT, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Elektrik-Elektronik Mühendisliği / Electrical and Electronics Engineering'),  
 ),  
 body: ListView.builder(  
 itemCount: huseyinHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(huseyinHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 );  
 }  
}  
  
class GeomaticsEngineeringPage extends StatelessWidget {  
 List<String> ermanHocaLessons = [  
 'Şube3: Dr. Öğr. Üyesi Erman ÜLKER (40/40)',  
 'Açıklama:',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube5: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#5: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Harita Mühendisliği / Geomatics Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: ermanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(ermanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class CivilEngineeringPage extends StatelessWidget {  
 List<String> ermanHocaLessons = [  
 'Şube2: Dr. Öğr. Üyesi Erman ÜLKER (40/40)',  
 'Açıklama:',  
 'Grup#2: Dr. Öğr. Üyesi Erman ÜLKER, Sondaj Akışkanlarının Reolojik Özelliklerinin Belirlenmesi ve Sirkülasyon sistemindeki Basınç Kayıplarının Hesaplanması',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube4: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#4: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('İnşaat Mühendisliği / Civil Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: ermanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(ermanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class MechanicalEngineeringPage extends StatelessWidget {  
 List<String> ebubekirHocaLessons = [  
 'Şube6: Dr. Öğr. Üyesi Ebubekir ATAN (40/40)',  
 'Açıklama:',  
 'Grup#6: Dr. Öğr. Üyesi Ebubekir ATAN, Mühendislik Uygulamaları için Metalurjik Atıklardan Sürdürülebilir Malzemelerin Tasarımı, Üretimi ve Karakterizasyonu',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube8: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#8: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Makine Mühendisliği / Mechanical Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: ebubekirHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(ebubekirHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class MechatronicsEngineeringPage extends StatelessWidget {  
 List<String> erkinHocaLessons = [  
 'Şube2: Doç. Dr. Erkin GEZGİN (40/40)',  
 'Açıklama:',  
 'Grup#2: Doç. Dr. Erkin GEZGİN, Fiziksel İnsan Robot Etkileşimi İçin İstek Tanımlama ve Robot Manipülatörler Üzerinde Uygulanması',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube4: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#4: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Mekatronik Mühendisliği / Mechatronics Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: erkinHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(erkinHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class MetalurgicalEngineeringPage extends StatelessWidget {  
 List<String> mucahitHocaLessons = [  
 'Şube5: Dr. Öğr. Üyesi Mücahit SÜTÇÜ (40/40)',  
 'Açıklama:',  
 'Grup#5: Dr. Öğr. Üyesi Mücahit SÜTÇÜ, Mühendislik Uygulamaları için Metalurjik Atıklardan Sürdürülebilir Malzemelerin Tasarımı, Üretimi ve Karakterizasyonu',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube7: Dr. Öğr. Üyesi Serkan KARAKIŞ (40/40)',  
 'Açıklama:',  
 'Grup#7: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Metalurji ve Malzeme Mühendisliği / Metalurgical and Material Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: mucahitHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(mucahitHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class PetroleumEngineeringPage extends StatelessWidget {  
 List<String> canHocaLessons = [  
 'Şube 5 (Doç. Dr. Can POLAT) (40/40)',  
 'Açıklama:',  
 'Grup#5: Doç. Dr. Can POLAT, Sondaj Akışkanlarının Reolojik Özelliklerinin Belirlenmesi ve Sirkülasyon sistemindeki Basınç Kayıplarının Hesaplanması',  
 ];  
  
 List<String> serkanHocaLessons = [  
 'Şube 7 (Dr. Öğr. Üyesi Serkan KARAKIŞ) (40/40)',  
 'Açıklama:',  
 'Grup#7: Dr. Öğr. Üyesi Serkan KARAKIŞ, Sensörler ve programlama kullanılarak mesafe ve yön tayini',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Petrol ve Doğalgaz Mühendisliği / Petroleum and Natural Gas Engineering'),  
 ),  
 body: Column(  
 children: [  
 ListTile(  
 title: Text('1.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: canHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(canHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ListTile(  
 title: Text('2.Seçenek'),  
 ),  
 ListView.builder(  
 shrinkWrap: true,  
 itemCount: serkanHocaLessons.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(serkanHocaLessons[index]),  
 onTap: () {Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => FinalPage(),  
 ),  
 );  
 //  
 },  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}  
  
class FinalPage extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Final Page'),  
 ),  
 body: Center(  
 child: Text(  
 'Tebrikler, Ders Seçimini Tamamladınız.',  
 style: TextStyle(fontSize: 20),  
 ),  
 ),  
 );  
 }  
}

Uygulamada geliştirilmeye devam edilecek olan kısımlar mevcuttur; bunlardan ilk olarak giriş (login) ekranı değiştirilecektir. Giriş ekranında öğrencilerden şifre ile kaydolmaları istenecek ve bu bilgiler Database ile bağlanarak tüm bilgiler kayıt altına alınacaktır. İkinci bir konu ise, uygulamanın daha sonrasında kontrol edilmesi için, admin paneli eklenecek ve farklı kullanıcılara admin yetkileri verilecektir. Bu sayede, yeni derslerin eklenmesi, kontenjan sayısının belirlenmesi gibi işlemler adminler tarafından kontrol altında olacaktır. Diğer bir konu ise okulumuzun renkleri olan bordo ve beyaz renkleri ile uygulamanın içindeki renkler tekrar düzenlenecek ve mobil uygulama okulumuzun renklerine uygun bir hale getirilecektir. Mobil uygulamamızın en önemli kısmı olan GPA kısımının detaylıca testleri yapılacaktır. Uygulamamızın son ekranındaki final ekranı değiştirilecek, kullanıcılara daha detaylı bilgiler iletilecektir. En son olarak ise, buradaki özellik de önemli bir uygulama özelliği olan; mail adresinden kayıt yapıldıktan sonra kullanıcı ders seçme işlemini uygulamadan tamamen tamamlandığında, geri bir bildirim ile kullanıcılara seçtiği ders ile ilgili bir mail iletilecektir.

GPA’in kod kısmında kontrol edildiği kısım aşağıdaki şekildedir;

import 'package:flutter/material.dart';

void main() {

runApp(LoginScreen());

}

class Student {

String firstName;

String lastName;

String email;

double gpa;

Student(this.firstName, this.lastName, this.email, this.gpa);

}

class LoginScreen extends StatelessWidget {

List<Student> students = [

//

];

List<Student> sortedStudents = [];

List<double> sortedGPAs = [];

LoginScreen() {

sortedStudents = List.from(students)

..sort((a, b) => b.gpa.compareTo(a.gpa));

sortedGPAs = List.from(sortedStudents.map((student) => student.gpa))

..sort((a, b) => b.compareTo(a));

}

Sonuç

Sonuç olarak, İKÇU bünyesindeki öğrencilerin kullanabileceği, ders seçmelerinde gerekli olan katkı ve faydayı en hızlı, güvenilir ve doğru bir şekilde ders seçmelerine yardımcı olacak, güncel bir mobil uygulama yapılması amaçlanmıştır. İleriki dönemlerde de belirtilen mobil uygulama güncel olan ihtiyaçlar doğrultusunda, öğrenciler ve sizlerden gelen tüm geri bildirimlerle güncellenmeye devam edilecektir. Burada amaçlanan ana hedef; İKÇU öğrencilerinin ders seçmesi için kullanabileceği bir modern uygulama yapılması amaçlanmıştır.

Kaynaklar

<https://docs.google.com/forms/d/1biG1MZ1xC2Wj8BwFw6wNcVJ5wdUeICDNlBzJpnlkDrI/edit?pli=1>

<https://muh.ikcu.edu.tr/S/19884/duyurular-eng401>

pub.dev

flutter.dev

<https://stackoverflow.com>

<https://www.geeksforgeeks.org>

Özgeçmiş

Adı Soyadı: Yiğit BAL

Eğitim: 2022–2023 İzmir Kâtip Çelebi Üniversitesi, Yazılım Müh. Bölümü

İş Deneyimi: 2019 – Etiya Bilgi Teknolojileri Kıdemli Yazılım Destek Uzmanı