



SAKARYA
ÜNİVERSİTESİ

T.C

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 - BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

GÜNLÜK ALGORİTMA ÖNERİ PROGRAMI

ALGOCHALLENGE

B181210382 – Ebru TERZİOĞLU

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Danışman : Prof. Dr. Ümit KOCABIÇAK

2020 – 2021 Güz Dönemi

T.C
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 - BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

B181210382 – Ebru TERZİOĞLU

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / Tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....

Jüri Başkanı

.....

Üye

.....

Üye

ÖNSÖZ

Gün geçtikçe masaüstü veya dizüstü bilgisayarlar popülerliğini cep telefonları ve tabletlere bırakmakta. Bunun en büyük sebebi insanların teknolojiyi her an her yerde yanında kolayca taşıma isteği. Sıkça tercih edilen bu mobil cihazlar araştırmalara göre her 6 dakikada bir kullanıcı tarafından kontrol edilmektedir. Bu durumda bize gösteriyor ki mobil cihazların hayatımızda çok büyük bir rolü var.

Bende çalışmamda bu konunun gelişime hala açık bir konu olduğunu düşündüğüm için projemi bir mobil uygulama üzerine yapmaya karar verdim. Hala yazılım alanında kendimi geliştirmeye çalıştığım içinde bu alanda insanların kendini geliştireceği bir uygulama yapmaya karar verdim.

Geliştirmeye karar verdiğim mobil uygulama yazılım alanında kendini geliştirmeye karar vermiş insanların algoritma mantığını öğrenmesi ve günlük algoritma örnekleri ile pratik yapmasını sağlayan bir programdır.

Tasarım çalışmamda Danışman Hocam Prof. Dr. Ümit KOCABIÇAK'a göstermiş olduğu danışmanlıktan ve bize aktardığı bilgi birikiminden dolayı ve tasarımımda bana fikir ve destek veren Elif KADI'ya teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER	ii
ŞEKİLLER LİSTESİ	iv
ÖZET	v
1. BÖLÜM GİRİŞ.....	1
2. BÖLÜM GENEL KISIMLAR	2
2.1. Projenin Konusu	2
2.2. Projenin Anlaşılması	2
2.2.1. Projenin yapılma gereği ve amacı.....	2
2.2.2. Projenin seçilme nedeni	3
2.3. Daha Önce Yapılmış Çalışmalar	3
2.4. React-Native.....	3
2.4.1. Neden React Native	4
2.4.2. React Native Avantajları	4
2.5. Node.js.....	4
2.5.1. Node.js avantajları	5
2.6. Firebase	5
2.6.1 Firebase Avantajları.....	6
2.7. Android Studio	6
2.8. Expo	7
3. BÖLÜM UYGULAMA	8
3.1. AlgoChallenge Sayfalar	8

3.2. Cmd'ye Eklenen Komutlar.....	13
3.3. Projeye Firebase Entegre Etme	13
3.4. Authentication İşlemleri.....	14
3.5. Firebase İle Veritabanı İşlemleri	17
3.6. Görsel Düzenlemeler.....	19
4. BÖLÜM TARTIŞMA VE SONUÇ	21
KAYNAKLAR	22
ÖZGEÇMİŞ.....	23
BSM 401 TASARIM ÇALIŞMASI	24
DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI	24

ŞEKİLLER LİSTESİ

ŞEKİL 1 REACT NATİVE.....	4
ŞEKİL 2 NODE.JS	5
ŞEKİL 3 FIREBASE	6
ŞEKİL 4 ANDROID STUDIO	6
ŞEKİL 5 EXPO.....	7
ŞEKİL 6 STACK NAVİGATOR SAYFALAR	8
ŞEKİL 7 TAB NAVİGATOR SAYFALAR	9
ŞEKİL 8 STACK NAVİGATOR METODU	9
ŞEKİL 9 TAB NAVİGATOR METODU	10
ŞEKİL 10 PROFİL EKRANI.....	10
ŞEKİL 11 ALGORİTMA EKRANI.....	11
ŞEKİL 12 COMPİLER EKRANI.....	12
ŞEKİL 13 WEBVIEW KULLANIMI	12
ŞEKİL 14 FIREBASE PROJE	13
ŞEKİL 15 FIREBASE CONFİG	14
ŞEKİL 16 GİRİŞ VE KAYIT EKRANI	15
ŞEKİL 17 HANDLESIGNUP METODU	16
ŞEKİL 18 HANDLELOGIN METODU	16
ŞEKİL 19 AUTHENTICATION EKRANI.....	17
ŞEKİL 20 VERİ TABANI EKRANI	18
ŞEKİL 21 FIRESTORE EKRANI.....	19
ŞEKİL 22 COLOR.JS DOSYASI	20

ÖZET

Anahtar kelimeler: Mobil Uygulama, React-Native, Firebase

Tasarımımın ana konusu insanların algoritma becerilerini geliştirmek yeteneklerini geliştirmek için bilgisayara gereksinim duymamalarını sağlamak ve onlara günlük derecesi giderek artan algoritmalar sunarak bu işin içine yavaşça girmelerini sağlamak.

Tasarımım olan uygulamanın amacı bu işi öğrenmek isteyen insanları boş zamanlarını değerlendirmek bilgisayar başında olmasalar dahi kendilerine sunulan algoritmaları çözebilmelerini sağlamak.

Uygulamamı React-Native ve Firebase yardımı ile gerçekleştirdim. React-Native kısaca Facebook tarafından üretilen, cross-platform mobil uygulama geliştirme olanağı sağlayan bir framework'tür. Cross-platform birden çok mobil işletim sistemi ile uyumlu uygulama yazılımlarının oluşturulmasıdır. Ben uygulamamı imkanlarım doğrultusunda android tabanlı gerçekledim. Firebase ise verilerimizi uzak bir sunucuda gerçek zamanlı olarak depolayabildiğimiz, bulut tabanlı bir platformdur.

1. BÖLÜM GİRİŞ

Günümüzde insanlar boş zamanlarını değerlendirebilmek ve eğlenmek amaçlı çeşitli türde uygulama ve oyunlar ile ilgilenmektedir. Gelişen teknoloji ile yapılan oyun ve uygulamalar kendini geliştirmiş ve insanlığın yararına birçok uygulama marketlerde yerini almıştır.

Genel olarak insanlar gün içinde zamanı güzel ve verimli geçirmek ister. Fakat bazı zamanlar vardır ki bir aktivite yapamayacak kadar kısa vaktimiz vardır. Bu duruma örnekler olarak; herhangi bir yere gidip gelirken dolmuş/otobüs beklerken geçirilmeye çalışılan zaman, yemek pişmesini beklerken, herhangi bir banka/alışveriş kuyruğunda sıra beklerken veya herhangi bir arkadaşımızı beklerken geçen zaman gibi örnekler gösterilebilir.

İnsanların çoğunun boş zamanlarını değerlendirmek amacı ile oyun oynadığı gözlemleniyor. Bu oyunların büyük bir kısmı mobil oyunlardır. Bunun en büyük nedeni mobil cihazların her an yanımızda taşıyabileceğimiz kadar küçük ve kullanışlı olmasıdır. Bu da artık insanların günlük hayatta sahip oldukları küçük zaman dilimlerini oyun oynayarak veya telefonlarında haber/dergi/kitap okuyarak geçirebilmelerini sağlamıştır.

Mobil oyun/uygulama geliştirmek için birçok ücretli ücretsiz platform bulunmaktadır ve bu platformlarda birçok oyun/uygulama geliştirilmiştir ve mağazalarda yayınlanarak kullanıcının beğenisine sunulmuştur. Mobil uygulama ve oyun geliştirme platformlarından en popüler olanları; Unity, Unreal Engine, React Native, Android Studio, Flutter olarak gösterilebilir. 2020'nin üçüncü çeyreğinin sonundaki sonuçlara baktığımızda en büyük mobil mağazalardan biri olan Google Play Store'da yaklaşık 2,7 milyon mobil uygulama bulunmaktadır. Büyük mağazalardan bir diğeri olan App Store da ise 1,82 milyon uygulama bulunmaktadır.

Günümüz koşullarında pandeminin de etkisi ile 2020'nin üçüncü çeyreğinde Google Play Store'da indirme sayıları 28,3 milyar gibi bir rakam gördü. Baktığımız zaman bu rekor bir sayı ve insanların kendilerini oyalayacak uğraşlar aradığının bir işareti idi. 2020'nin ikinci çeyreğinde oyun uygulamalarının Google Play Store'da en popüler kategori olduğu görülüyor. Oyunlar uygulamalar arasında yüzde 13,49'luk bir dilimi kapsıyor. Eğitim amaçlı uygulamalar, yüzde 9,11'lik bir dilim ile ikinci sırayı alırken üçüncü dördüncü sırada "iş" ve "araç" uygulamaları yer alıyor[1].

2. BÖLÜM GENEL KISIMLAR

Günümüzde mobil cihaz ve uygulamaların kullanımı gün geçtikçe artmaktadır. Bunun ile birlikte teknoloji de o yönde gelişim göstermekte ve o yönde ürünler öne sürülmektedir. Kullanıcılar her an her özelliğin yanlarında olmasını ve bir tık ile işlemlerini gerçekleştirebilmek istemektedir. Hatta kod yazmak için bile bu durum geçerli hale gelmiştir. Kullanıcılar bilgisayar başında olmasalar dahi o an çözmeleri gereken algoritmaları mobil compiler'lar yardımı ile o an çözüme kavuşturmak istemektedir. Bende bu alandaki ihtiyaçları gidermek adına bir mobil uygulama geliştirmeye karar verdim.

2.1. Projenin Konusu

Tasarımımın ana konusu algoritma becerilerini geliştirmek isteyen insanların her an her yerde müsait olduğu her an bunu yapabilmelerini sağlamak. Geliştirdiğim uygulamada her gün zorluk seviyesi artan algoritmalar kullanıcıya sunulmaktadır. Algoritmaları mobile entegre edilmiş compiler tarafından kullanıcı çözebilmektedir. Buna göre bir puan alıp uygulamayı kullanan kullanıcılar arasındaki sıralamasını da görebilmektedir kullanıcı.

2.2. Projenin Anlaşılması

2.2.1. Projenin yapılma gereği ve amacı

Tasarımımın yapılma amacı kodlama ve algoritma geliştirme konusunda fırsat eşitliğinin sağlanması için bir adım daha ilerlenebilmesi.

Bu proje ile kullanıcılar ellerinde bir bilgisayar olsa da olmasa da günlük verilen algoritma örneklerine ulaşip aylık challenge'lara katılma hakkı kazanabilecek. Yazılımı angarya bir iş gibi değil de bir oyun gibi göstermek ana amaçtır uygulamada. Bu işin içinde olan ve hali hazırda bilgi birikimi bulunan insanların günlük olarak başlangıçta belirtilen seviyeye göre gönderilen algoritmalar ile algoritma becerilerini daha da geliştirecekleri bir ortam sunmak amaçlanmıştır.

2.2.2. Projenin seçilme nedeni

Bende hali hazırda yazılım alanında kendimi geliştiriyorum ve algoritma çözme becerisinin yazılımda önemini farkındayım. Bu beceriyi geliştirmek için sürekli olarak alıştırma yapmak gerektiğinin de fazlası ile farkındayım. Her zaman alıştırma yapmak için bilgisayar başında olamıyorum ve bunun bir eksik olduğunu düşünüyorum. İstedğim her an istediğim her yerde kütüce ağır bilgisayarımı taşıma gereği hissetmeden bu alıştırmaları yapabilmek için böyle bir uygulama geliştirmeye karar verdim.

2.3. Daha Önce Yapılmış Çalışmalar

Encode : Learn to Code, Lightbot : Code Hour, UdaCity – Learn Programming[5]

Yukarda belirttiğim uygulamalar daha kapsamlı kodlamayı baştan sona öğreten uygulamalar benim tasarladığım uygulama ise daha çok algoritma çözme pratiği yapmak için basit bir arayüz ile hazırlanmış bir uygulama olarak tasarlandı. Daha çok bu işin başında olan insanları bu işten soğutmadan gözlerini korkutmadan onları işin içine çekmek ana amaçtır. Tasarım fikri bulunurken Hackerrank adlı siteden esinlenilmiştir.

2.4. React-Native

Bir kere öğren, her yerde yaz.

React Native, bir kullanıcı ara yüzü geliştirme kütüphanesi olan React ile cross-platform uygulama geliştirdiğiniz bir framework. Ama burada üzerinde durmamız gereken asıl kelime React değil Native kelimesidir. Bir React-Native uygulaması HTML ile oluşturduğumuz bir hybrid uygulama değildir, webview içerisinde çalışmaz.

React Native Facebook tarafından geliştirilen bir front-end frameworktür. Tek bir dil üzerinden geliştirilir. Aynı anda hem IOS hem Android işletim sistemlerinde mobil uygulama geliştirme imkânı sağlar.



Şekil 1 React Native

2.4.1. Neden React Native

React-Native yayınlandığı günden itibaren sürekli bir ivme kazanarak geliştirici dostu ve geliştiricilerin en popüler uygulama geliştirme platformu haline gelmiştir.

React-Native'in bu çok tercih edilmesi karşılaştığım ve karşılaşacağım sorunlarını çözümüne daha kolay ulaşmamı sağladı. Çoğu yazılım geliştiricinin daha önce bildiği veya aşina olduğu teknolojileri (HTML, CSS, Javascript vb) içinde barındırıyor[2].

2.4.2. React Native Avantajları

Tek bir platform üzerinden hem Android hem IOS platformuna uygun uygulamalar geliştirme imkânı sağlıyor[3].

Uygulama geliştirme hızı diğer platformlara kıyasla çok daha fazla. Yapılan değişiklik kaydedildiği anda uygulamaya yansıyor.

Platform çok fazla kişi tarafından kullanıldığı için içerisindeki hata ve optimizasyon problemleri hızlı bir şekilde çözüme kavuşuyor. Ayrıca yine çok fazla kişinin kullanması bizim karşılaştığımız hatalar ile yüz yüze gelme ihtimali olan kullanıcı sayısını artırıyor ve çözüme daha hızlı ilerlememizi sağlıyor. Ayrıca dokümantasyonu çok temiz ve anlaşılır.

2.5. Node.js

Node.js, Javascript ile server taraflı uygulamalar yazabileceğimiz, Joyend tarafından 2009 yılında geliştirilmeye başlanmış bir Javascript runtime platformudur. V8 isimli bir Javascript motoru üzerinde çalışır.



Şekil 2 Node.js

2.5.1. Node.js avantajları

Olay tarafından yönlendirilen ve bloke olmayan özellikleri nedeni ile eş zamanlı bağlantıları rakiplerine göre çok daha hızlı ve etkin şekilde işler.

Hem istemci (client) hem de sunucu (server) tarafında aynı dili kullanabilmenin büyük kolaylığını kullanıcılara sunar. Açık, etkin API kullanımına olanak sağlar.

Bloke olmama durumu : Node.js aynı anda çok sayıda istem alabilir. Node programlama modelinde her şey asenkrondur; ancak Node.js çekirdeğinde pek çok fonksiyonun hem senkron hem de asenkron sürümleri mevcuttur. Node.js'in avantajlarından yararlanmak için asenkron sürümler tercih edilmelidir[4].

2.6. Firebase

Firebase Google tarafından sunulan mobil ve web uygulaması geliştiren yazılımcıların hızlı geliştirme yapmasını sağlayan, işlerini aşırı derecede kolaylaştıran birçok ürün ve servis sunan bir back-end servistir.

Uygulamalara anlık olarak hizmet veren bir bulut alt yapısıdır. Back-end işlemleri yapmadan SDK sayesinde uygulamalara veri gönderebilir, alınabilir, güncellenebilir ya da silinebilir. Firebase NoSQL'dir. İnternet bağlantısı olmadan kullanamayız.

2.6.1 Firebase Avantajları

Firebase kullanıcılara analitik, veri tabanları ve kilitlenme raporları gibi işlevler sunar, böylece hızlı hareket edebilir ve kullanıcılarınıza odaklanabilirsiniz. Google alt yapısı üzerine kuruludur ve en büyük uygulamalar için bile otomatik olarak ölçeklenir.



Şekil 3 Firebase

2.7. Android Studio

Android Studio, Android uygulamalarının geliştirildiği, üst seviye özelliklere sahip ve Google tarafından da önerilen resmi programlama aracıdır.



Şekil 4 Android Studio

2.8. Expo

Expo, XCode veya Android Sturdio kurulumu yapmaya gerek duymadan bir projeyi başlatmamıza olanak sağlar. Uygulamayı direk telefonumuza aktarabilmemiz en büyük avantajlarındanr. İnternetsiz çalışmaz.



Şekil 5 Expo

3. BÖLÜM UYGULAMA

Uygulamamda giriş ekranından sonra profil ekranınız görünüyor. Bu ekranda çözdüğünüz algoritma sayısı ve puanınız yer alıyor. Ayrıca uygulamayı kullanan kişiler arasındaki sıralamanızı da burada görebilirsiniz. Diğer ekranlarda günün algoritması ve çözümü yazacağımız kutucuk yer alıyor. Üçüncü ekranda ise compiler'ımız yer alıyor. Projeyi gerçeklerken önce android emulator üzerinden gerçeklemeye çalıştım ama yaşadığım sorunların çözümüne ulaşamamam, çoğunluğun expo tercih etmesi ve Expo'da bireysel sorunlarıma daha kolay çözüm bulabilmem kullandığım platformu expo olarak değiştirmeme neden oldu.

3.1. AlgoChallenge Sayfalar

Kullanıcı uygulamaya girdiği anda kendisini giriş ekranı karşılar. Eğer hesabı varsa bu ekran üzerinden kolayca giriş yapabilir. Eğer bir hesabı yoksa giriş yap butonunun hemen altında bulunan hesap oluştur butonuna tıklayarak. Kayıt sayfasına yönlendirilir. Kayıt olduktan veya giriş yaptıktan sonra ana sayfaya yönlendirilir. Ama sayfada ise karşımıza 3 adet buton çıkar sayfalar arası geçişi sağlamak için. Bu sayfalar arasındaki geçiş @react-navigation/native ile sağlanmaktadır.

@react-navigation/native paketi yüklendikten sonra geçiş yapmak istediğim sayfaları aşağıdaki gibi oluşturdum. Oluşturduğum bu sayfaları direk App.js dosyası içinde kullanmadım. AppContainer adlı bir js dosyası içinde oluşturdum.

```
import SignUp from "../screens/SignUp";
import SignIn from "../screens/SignIn";
import ForgatPasword from "../screens/ForgatPasword";
import MainPage from "../screens/MainPage";
```

Şekil 6 Stack Navigator Sayfalar

Burada tanımlanan MainPage sayfası bizim ana sayfamız. Anasayfada tablar arası geçişi sağlayan TabNavigator'ımız burada yer alıyor. Aşağıdaki şekildeki gibi tablarımızı MainPage içine import ettik.

```
import Profile from '../screens/Profile';  
import Algorithm from '../screens/Algorithm';  
import Compiler from '../screens/Compiler';
```

Şekil 7 Tab Navigator Sayfalar

Stack navigator sayfalarını oluşturmak için createStackNavigator() metodunu, Tan navigator sayfalarını oluşturmak için ise createBottomTabNavigator() metodunu kullandım.

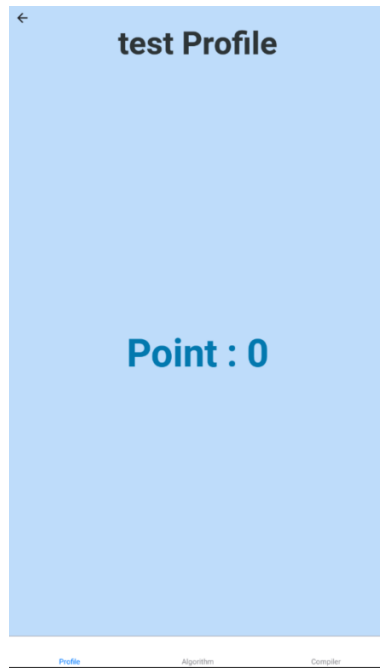
```
const Stack = createStackNavigator();
```

Şekil 8 Stack Navigator Metodu


```
const TabNavigator = createBottomTabNavigator({
  Profile: {screen: ProfileScreen},
  Algorithm: {screen: AlgorithmScreen},
  Compiler: {screen: CompilerScreen},
});
```

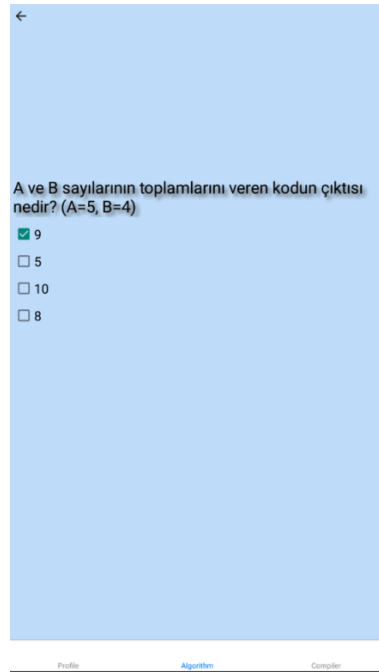
Şekil 9 Tab Navigator Metodu

Yukarıda da görüldüğü gibi Ana sayfamda 3 ayrı tab bulunmakta. Birinci tab ana sayfa açıldığı gibi karşımıza çıkan tab'dır. Burada kullanıcının score'u, çözdüğü algoritma sayısı gibi bilgiler yer almaktadır. İkinci sayfada günü algoritması bulunur ve altında işaretlenmesi için cevaplar yer alır. 3. Tabda ise compiler bulunur. Burada da algoritmayı çözmek için gerekli kodu yazıp çıktıyı kontrol edebiliriz.



Şekil 10 Profil Ekranı

Profil ekranın bizi Firebase'den çektiğimiz isim ve score bilgileri bulunuyor.



Şekil 11 Algoritma Ekranı

Algoritma sayfasında günlük olarak json dosyasından çekeceğimiz algoritmalar bulunuyor. Altta da checkboxlar kullandık. Checkbox kullanabilmek için `@react-native-community/checkbox`'ı projemize indirmemiz ve import etmemiz gerekiyor.

```
import CheckBox from "@react-native-community/checkbox";
```



Şekil 12 Compiler Ekranı

Compiler ekranı için bir compiler yazmaktansa açık kaynak kodlu bir sayfaya webview yardımı ile yönlendirme yapıyoruz. Bunun için önce react-native-webview I projemize kuruyoruz. Daha sonrasında import ediyoruz.

```
import { WebView } from "react-native-webview";
```

```

class Compiler extends Component {
  render() {
    return (
      <>
        <View style={styles.container}>
          <WebView
            source={{ uri: "https://www.onlinegdb.com/online_c++_compiler" }}
          />
        </View>
      </>
    );
  }
}

```

Şekil 13 Webview Kullanımı

3.2. Cmd'ye Eklenen Komutlar

expo init ProjeAdi // Proje dosyasını oluşturmak için

cd ProjeAdi // Proje dizinine geçmek için.

npm install --save react-navigation // Sayfalar arası geçiş için.

npm install --save react-native-gesture-handler // Gesture işaretler için eklenen native bağımlılığı olan paket.

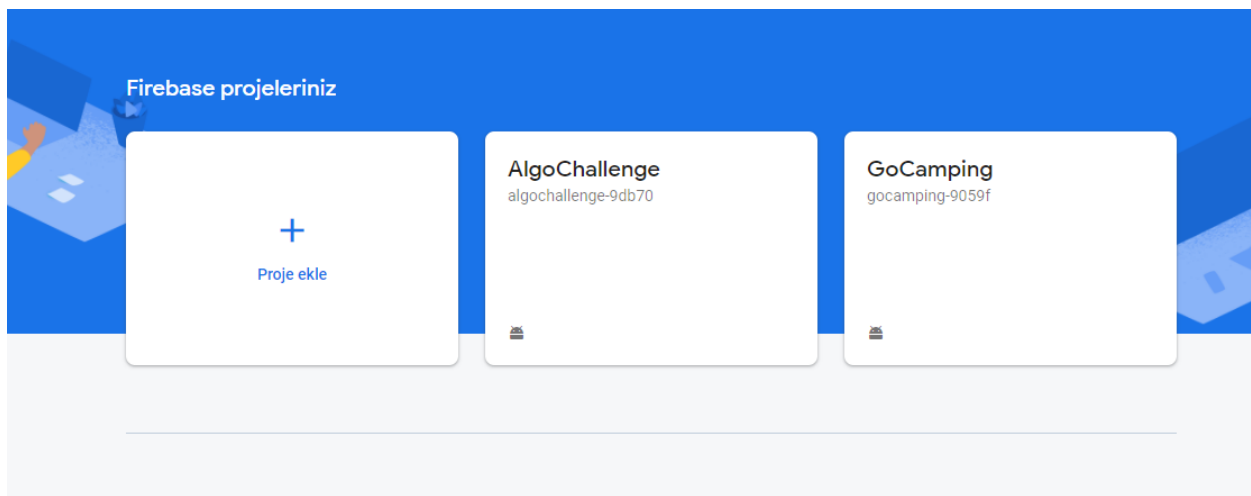
react-native link react-native-gesture-handler // Native bağımlılığı olduğundan link işlemi yapılır

3.3. Projeye Firebase Entegre Etme

Cmd ekranını açarak proje klasörünün kök dizinine giriyoruz. Girdiğimiz proje klasöründe aşağıdaki komutu çalıştırıyoruz.

npm install firebase --save // firebase yükleyebilmek için.

Firebase sitesini açıp yeni bir proje oluşturuyoruz. Bu oluşturduğumuz projede kullanıcı bilgilerini tutacağız. AlgoChallenge isimli proje içinde verilerimiz saklanacaktır. GoCamping adlı uygulama tamamen kendi kişisel uygulamamdır.



Şekil 14 Firebase Proje

Proje adınızı girip gerekli koşulları kabul ettikten sonra oluşturulan Firebase'i uygulamanıza ekleyebilirsiniz.

Uygulamamda config klasörü içinde Firebase.js adında bir dosya oluşturdum. Bu dosyada Firebase için gerekli config değerlerini tutuyorum.

```
import firebase from "firebase";

// Firebase config ayarları
var firebaseConfig = {
  apiKey: "AIzaSyDTz-0CWVhbSZj6nZQfneocIHNJs1XhBwc",
  authDomain: "algochallenge-9db70.firebaseio.com",
  projectId: "algochallenge-9db70",
  databaseURL: "https://algochallenge-9db70-default-rtdb.firebaseio.com/",
  storageBucket: "gs://algochallenge-9db70.appspot.com",
  messagingSenderId: "891102170026",
  appId: "1:891102170026:android:9566e007ff8f8c5d3479d3",
};

// Firebase Initialize ediliyor
let Firebase = firebase.initializeApp(firebaseConfig);

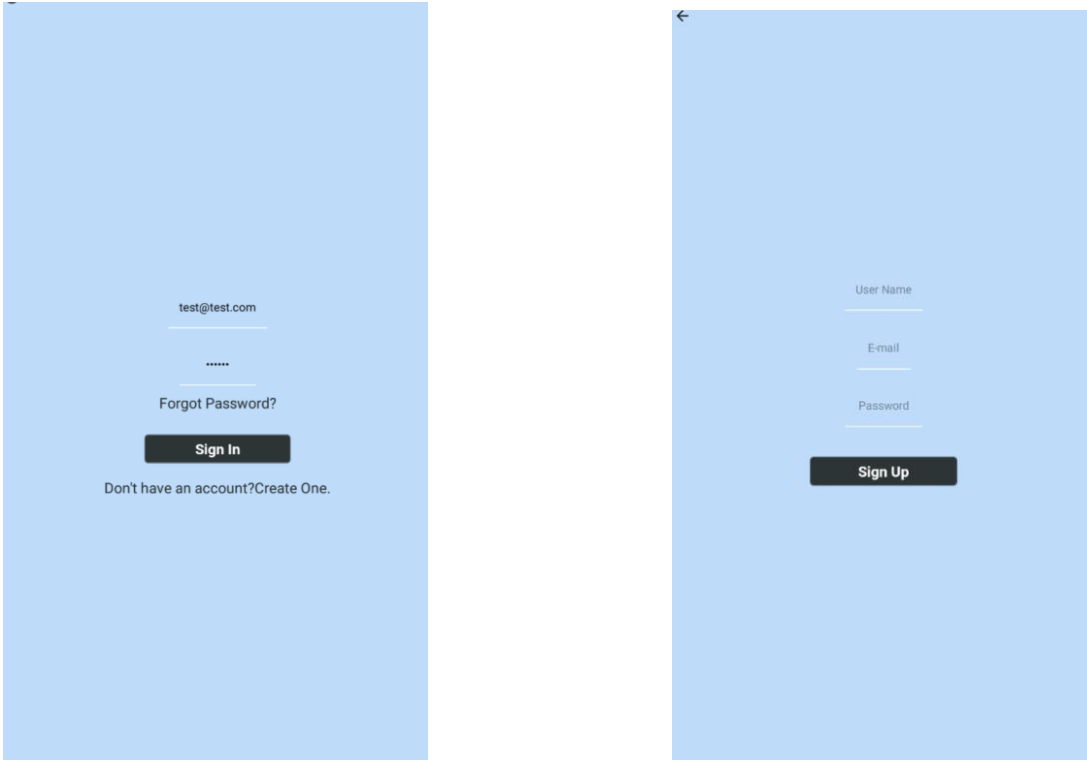
export default Firebase;
```

Şekil 15 Firebase Config

3.4. Authentication İşlemleri

Kullanıcıların e-posta ve şifre yolu ile giriş yapmalarını sağlamak için Firebase projemde authentication kısmını email ve şifre için etkinleştirdim.

Giriş ve Kayıt sayfalarımın tasarımlarını tamamladım.



Şekil 16 Giriş ve Kayıt Ekranı

Yeni kullanıcı oluşturmak için SignUp.js sayfasındaki handleSignUp metodunu ekliyoruz. Bu metod kullanıcının girmiş olduğu e-mail ve şifre bilgilerini alır, bu bilgiler ile firebase veri tabanına kullanıcı eklenir. Uygulamaya kayıt olunmuş olur. Kayıt tamamlandıktan sonra otomatik olarak ana sayfaya yönlendirilirsiniz.

Eğer e-mail veya şifre ile ilgili bir sorun olursa bunu catch ile yakalarız. Sorunun ne olduğu ile ilgili bilgiyi kullanıcıya vermek için ekrana hata mesajı yazdırılır. Bu hata mesajındaki hatayı fark eden kullanıcı gerekli düzeltmeyi yapabilir.

```

handleSignUp = () => {
  Firebase.firestore().collection("user").doc(email).set({
    Username: username,
    Score:0
  });
  Firebase.auth()
    .createUserWithEmailAndPassword(email, password)
    .then(() => navigation.navigate("MainPage", { caption: email }))
    .catch((error) => console.log(error));
};

```

Şekil 17 handleSignUp Metodu

Kullanıcının giriş yapabilmesi için SignIn.js sayfasında handleLogin metodunu ekliyoruz. Bu metod firebaseAuth.signInWithEmailAndPassword ile girilen email ve şifre kombinasyonunu veri tabanından kontrol eder.

Eğer email ve şifre bilgileri örtüşüyorsa sistem bizi otomatik olarak ana sayfaya yönlendirir. Email ya da şifre ile ilgili bir sorun var ise bu sorunu catch fonksiyonu ile yakalarız. Yakalanan sorun kullanıcının bilgilendirilmesi amacı ile ekrana uyarı olarak yazdırılır. Hata nedeni ile hatayı düzeltene kadar kullanıcı sisteme giriş yapamaz.

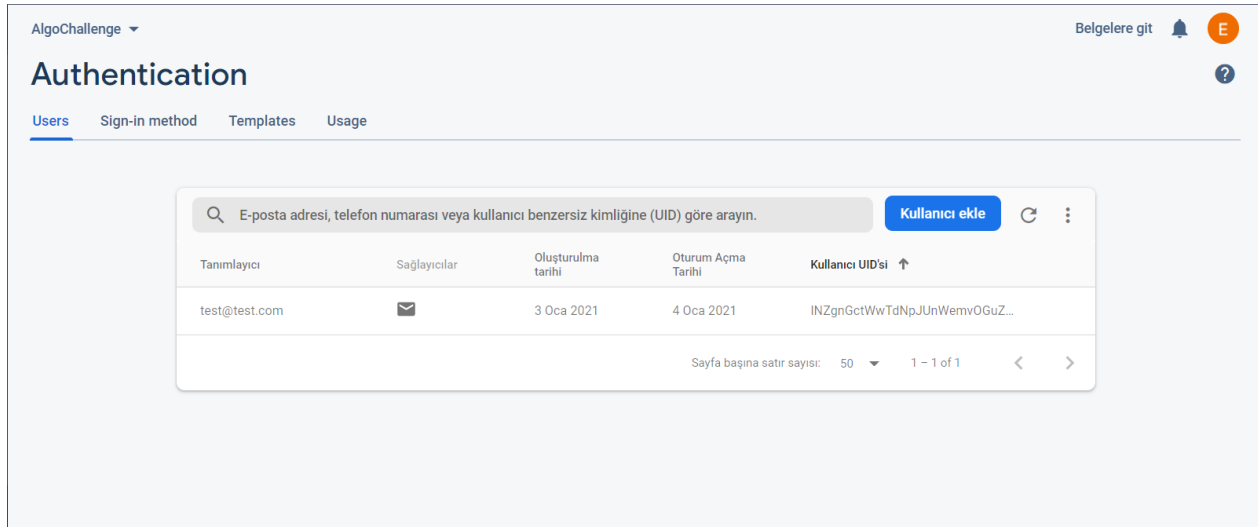
```

handleLogin = () => {
  Firebase.auth()
    .signInWithEmailAndPassword(email, password)
    .then(() => navigation.navigate("MainPage", { c: email }))
    .catch((error) => alert(error));
};

```

Şekil 18 HandleLogin Metodu

Aşağıda authentication işlemini gerçekleştirdiği sayfa ve test kayıtlı kişisi görünmektedir.



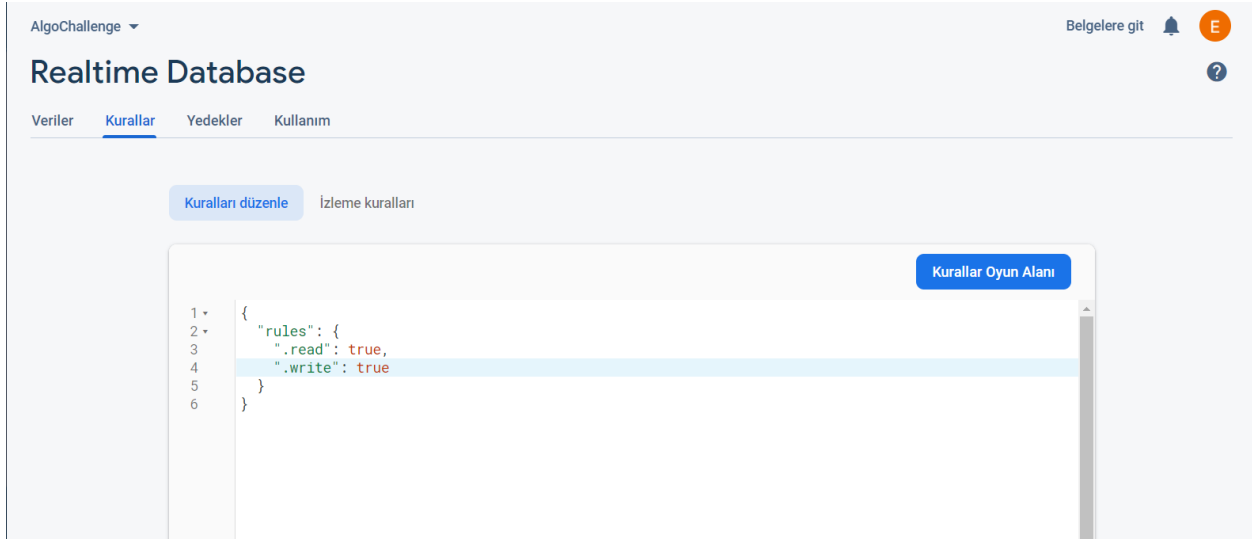
Şekil 19 Authentication Ekranı

3.5. Firebase İle Veritabanı İşlemleri

Firebase database'i kullanıcılara asenkron yapıda çalışan gerçek zamanlı bir NoSql veri tabanı hizmeti verir. Veri tabanındaki veriye uygulamadan ulaşabilmek için kimlik doğrulaması gerekir. Normalde web ortamında veya mobilde veri tabanı ile işlem yapmak için bir veri tabanı kurmak zorundayız. Oluşturduğumuz uygulamaya göre veri tabanına ulaşmak için servisler yazarız, o servisleri sunuculara push ederiz. Ama Firebase'in database servisi sayesinde basit bir şekilde bir veri tabanı uygulamaya dahil edip kullanabiliriz.

Firebase tanıdık bir veri tabanı yapısına sahip değildir. Veriler tablo biçiminde değil json objeleri şeklinde depolanır. Bu verileri daha kolay ekleyip silmemizi sağlar.

Oluşturulan veri tabanı projeye eklendikten sonra veri tabanınızı okunur ve yazılabilir yapmak gerekiyor. Bunun için Firebase'e gidip projemizin veri tabanını açtıktan sonra yandaki menüden Database'i seçmemiz gerekiyor. Buradan aynı zamanda verilerimizi görebiliriz.



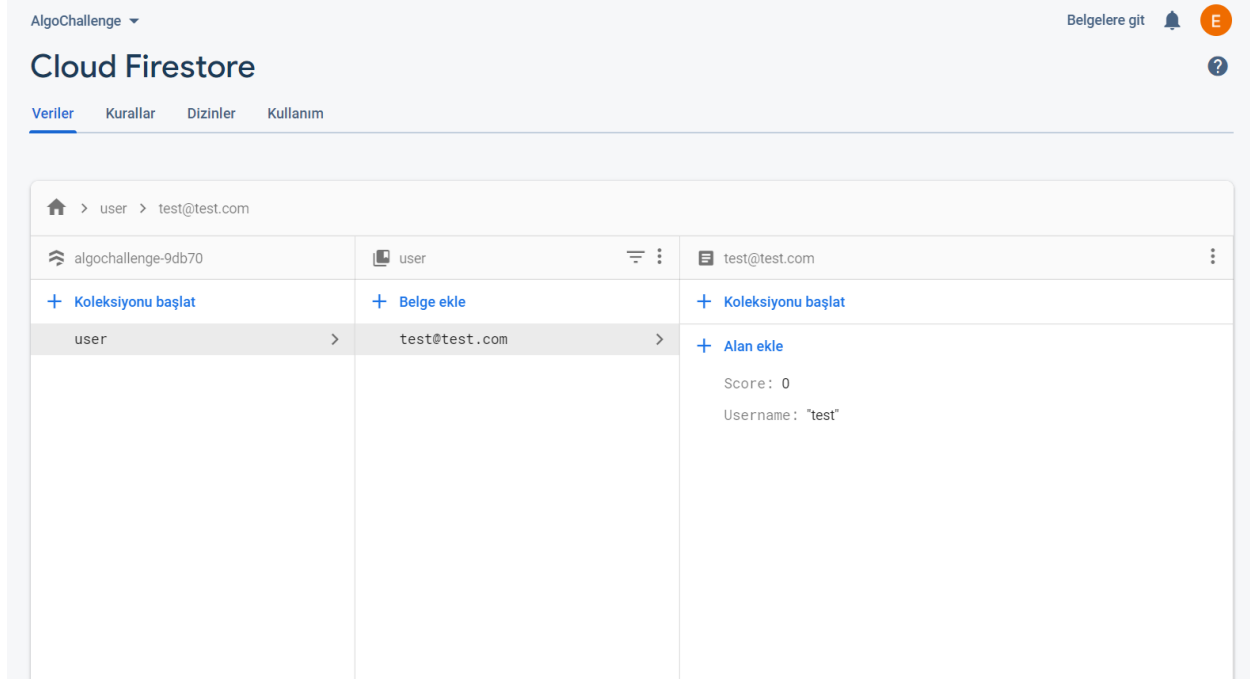
Şekil 20 Veri Tabanı Ekranı

Ben verilerimi Cloud Firestore üzerinde tutmaya karar verdim. Cloud Firestore Google'ın büyük ve karmaşık verileri saklamak ve hızlıca sorgulamak amacı ile geliştirdiği doküman tabanlı Nosql veri tabanıdır.

Firestore esnek bir yapıya sahiptir. Veriler istenilen şemada tutulabilir. Veriler doküman adı verilen yapılarda saklanır. Doküman topluluğuna ise Koleksiyon adı verilir. Bu yapısı itibari ile diğer Nosql veri tabanlarına benzemektedir.

Firestore'u RealTime Database'den ayıran özelliği daha karmaşık verileri saklamak ve sorgulamak için gelişmiş çözümler sunmasıdır. Yani iç içe verileri saklamak ve sorgulamak için Firestore daha performanslı, ölçeklenebilir ve esnek bir yapı sağlıyor.

Ben projemde score ve kullanıcı ismi gibi verileri Cloud Firestore üzerinde tuttum.



Şekil 21 Firestore Ekranı

3.6. Görsel Düzenlemeler

Projenin çalışabilirliğinin yanında görsel olarak kullanıcıya hitap etmesinin de önemli olduğunu düşünüyorum bu yüzden bazı düzenlemeler yapılması gerekti. Bu düzenlemeleri yaparken işimizi kolaylaştırmak için bazı işlemler yaptım. Örneğin renklerin her seferinde hex kodlarını aramak istemedim. Kullanmayı planladığımız renkleri araştırıp hex kodlarını topladık ve bir component içinde birleştirdim. Gereken dosyalarda bu oluşturduğumuz component'i çağırarak işlemlerimizi gerçekleştirdim

```
export default colors = {  
  black: "#2D3436",  
  blue: "#0278ae",  
  lightBlue: "#bedcfa",  
  white: "#FFFFFF",  
  green: "#70af85",  
  lightGreen: "#70af85",  
  pink: "#cd5d7d",  
  lightPink: "#fdcfdf",  
  gray: "#A4A4A4",  
  lightGray: "#CACACA"  
};
```

Şekil 22 Color.js Dosyası

Kullandığımız renkleri bu şekilde tanımlayıp sonrasında gereken yerlerde dosyaya import ederek çağırdık.

```
import colors from "../components/Colors";
```

4. BÖLÜM TARTIŞMA VE SONUÇ

Çalışmalarım sonucunda mobil uygulama geliştirirken nelere dikkat etmem gerektiğini, hazırlık aşamalarının ve ilerleyen sürecin nasıl geliştiğini görme fırsatı elde ettim ve bu yönde birçok bilgi edindim. Bunun için tasarladığım ve uyguladığım uygulama ile benzer nitelikte birçok profesyonel ve amatör uygulama inceledim. Kendi uygulamamı da nasıl tasarlayacağımı ve nasıl uygulayacağıma dair bu uygulamalar sayesinde fikir edindim.

Geliştirdiğim uygulama için gereken ortamlar hakkında daha detaylı bilgiler edindim ve kullanmak istediğim geliştirme ortamı olarak react native i uygun bulup onun üzerinde geliştirme yaptım. React Native i seçmemin en büyük nedeni cross platform bir uygulama olması. Yani hem IOS hem de Android işletim sistemine ekstra bir efora gerek kalmadan uygulama çıkabilmemiz beni React Native kullanmaya itti.

Geliştirdiğim uygulamanın temel halini prototipleşerek görseller ile sundum şu anda temel işlevleri yerine getirsen de daha ilerlenmesi gereken işlerinin olduğunun farkındayım. Henüz tasarım aşamasında olan uygulamamı ilerleyen süreçte daha kullanılabilir bir hale getirip daha da geliştireceğim.

KAYNAKLAR

1. <https://teknosafari.net/google-playin-indirmeleri-app-storedan-3-kat-daha-fazla/>
2. <https://medium.com/kodcular/react-native-nedir-7b333d319597>
3. <https://reactnative.dev/>
4. <https://www.argenova.com.tr/node-js-nedir-ne-icin-kullanilir#:~:text=Node.js%20JavaScript%20ile%20server,bir%20javascrip%20motoru%20%C3%BCzerinde%20%C3%A7al%C4%B1%C5%9F%C4%B1r.>
5. <https://webrazzi.com/2017/02/06/kodlama-ogreten-en-iyi-5-mobil-uygulama/>.

ÖZGEÇMİŞ

Ebru Terzioğlu, 24.09.1997’de Rize’de doğdu. İlkokulu Fındıklı’da, ortaokulu Artvin’de ve lise eğitimini Güneysu’da tamamladı. 2015 yılında Güneysu Şehit Kemal Mutlu Anadolu Öğretmen Lisesi’nden mezun oldu. 2015 yılında Sakarya Üniversitesi Elektrik Elektronik Mühendisliği Bölümü’nü kazandı. 2015 – 2016 yılları arasında hazırlık eğitimi aldı. 2017 yılında Aiesec ile gönüllük esaslı bir projede Ukrayna’ya giderek İngilizce öğretmenliği yaptı. 2018 yılında Elektrik Elektronik Mühendisliğini bırakıp yine Sakarya Üniversitesinde Bilgisayar Mühendisliği Bölümüne geçiş yaptı. Okul hayatında Sakarya Üniversitesi Enerji Teknolojileri Topluluğu, Sakarya Üniversitesi Robot Topluluğu, Sakarya Üniversitesi IEEE RAS gibi kulüplerde aktif üye olarak görev aldı. 2019 yılında Teknokent bünyesinde bulunan Omicron Games de yaz stajını yaptı ve o şirkette bir yıldan fazla bir süre çalıştı. Şu anda da hala Sakarya Üniversitesi Bilgisayar Mühendisliğinde öğrenimini sürdürmektedir. Aynı zamanda Brat Games adı altında Oyun yazılımcısı olarak çalışmaya devam etmektedir.

BSM 401 TASARIM ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU: AlgoChallenge

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): B181210382 / Ebru TERZİOĞLU

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA): Prof. Dr. Ümit KOCABIÇAK

DANIŞMAN İMZASI: