

HER YERDE EĞİTİM



2020

**BİLGİSAYAR MÜHENDİSLİĞİ
BİTİRME PROJESİ TEZİ**

Canan GÜNHAN

Zeynep Nur ŞEN

HER YERDE EĞİTİM

Canan Günhan

Zeynep Nur Şen

Karabük Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümünde

Bitirme Projesi Tezi

Olarak Hazırlanmıştır.

KARABÜK

Haziran 2020

Canan Günhan ve Zeynep Nur Şen tarafından hazırlanan “HER YERDE EĞİTİM”
başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Yasin ORTAKÇI

.....

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

...../...../2020

Bilgisayar Mühendisliği bölümü, bu tez ile, Bitirme Projesi Tezini onaylamıştır.

Dr. Öğr. Üyesi Hakan KUTUCU

.....

Bölüm Başkanı

“Bu projedeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımızı beyan ederiz.”

Canan GÜNHAN, Zeynep Nur ŞEN

ÖZET

Bitirme Projesi Tezi

HER YERDE EĞİTİM

Canan GÜNHAN

Zeynep Nur ŞEN

Karabük Üniversitesi

Bilgisayar Mühendisliği

Bilgisayar Mühendisliği Bölümü

Tez Danışmanı:

Dr. Öğr. Üyesi Yasin ORTAKÇI

Haziran 2020 , 56 Sayfa

Özel ders, eksik olunan bir konuda, konuya hakim bir kişinin yani eğitmenin o konuyu birebir olarak tamamlamak için verdiği eğitimidir. Teknolojinin gelişmesiyle internet üzerinden online eğitimler ne kadar yaygınlaşmış olsa da öğrenci için yeterli verimi sağlamamaktadır. Çünkü her ne kadar öğrenciler takıldıkları noktada mail ile bir geri dönüş alıyor olsa bile bu birebir eğitim kadar verimli olmamaktadır.

Doğru platform bulunmadığı için yetenekli ve alanına hakim eğitimcilerin verdiği özel dersler, yakın çevresinden uzaklaşmamaktadır. Tezimizin asıl amacı eğitimi yaygınlaştırmak ve özel derse istenilen kriterler doğrultusunda kolaylıkla ulaşmayı sağlamaktır.

Anahtar Kelimeler: Özel Ders, Birebir Eğitim

ABSTRACT

Senior Project Thesis

Education is Everywhere

Canan GÜNHAN

Zeynep Nur ŞEN

Karabük University

Faculty of Engineering

Department of Computer Engineering

Project Supervisor:

Asst. Prof. Dr. Yasin ORTAKÇI

June 2020, 56 pages

A private lesson is a training given by a person who is competent in the subject, ie the trainer, to complete the subject exactly. With the development of technology, online education over the internet has become widespread, but it does not provide sufficient efficiency for the student. Because even though students receive a return by mail at the point where they hang out, this is not as efficient as one-to-one education.

Since the right platform is not available, private lessons given by talented and educated educators cannot move away from their immediate surroundings. The main purpose of our thesis is to disseminate education and to provide easy access to private lessons in line with the desired criteria.

Keywords: Private Lesson, One-to-One Education

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde, oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığımız, yönlendirme ve bilgilendirmeleriyle çalışmamızı bilimsel temeller ışığında şekillendiren sayın hocamız Dr. Öğr. Üyesi Yasin ORTAKÇI'ya sonsuz teşekkürlerimizi sunarız.

İÇİNDEKİLER

KABUL	ii
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ŞEKİLLER DİZİNİ.....	ix
BÖLÜM 1.....	1
GİRİŞ.....	1
1.1. LİTERATÜR ÖZETİ.....	1
1.2. TEZİN AMACI.....	1
BÖLÜM 2.....	3
YÖNTEMLER, ARAÇLAR ve YAZILIMLAR.....	3
2.1. REST TEKNOLOJİSİ.....	3
2.2.RESTFUL API GELİŞTİRİLİRKEN KULLANILAN TEKNOLOJİLER...7	
2.2.1. PHPStorm.....	7
2.2.2. Mysql-PhpMyAdmin.....	9
2.2.3. Postman.....	10
2.3. RESTFUL API'nin TEST EDİLMESİ.....	11
2.4. ANDROİD MOBİL UYGULAMA GELİŞTİRİLMESİ.....	24
2.4.1.Android Mobil Uygulama Geliştirirken Kullanılan Teknolojiler.....	25
2.4.2. Android Studio.....	25
2.4.3. Android Studio'da Kullanılan Bileşenler.....	35
2.4.4. Mobil Uygulama Geliştirme Süreci.....	37

BÖLÜM 3

KAYNAKLAR.....	55
----------------	----

ŞEKİLLER DİZİNİ

Şekil 1. REST API'nin bütün programlama dilleri ile çalışması

Şekil 2. REST API'nin çalışma şekli

Şekil 3. RESTful API dizin yapısı

Şekil 4. RESTful API index.php

Şekil 5. Veritabanı tabloları

Şekil 6. RESTful API ile veritabanı bağlantısının yapılması

Şekil 7. Postman arayüzü ve Http metodları

Şekil 8. RESTful API ile öğrenci kayıt

Şekil 9. RESTful API ile öğretmen kayıt

Şekil 10. Veritabanında bulunan calenders tablosu

Şekil 11. RESTful API ile öğretmenin boş gün bilgisini güncellemesi

Şekil 12. RESTful API ile öğrencinin ders araması

Şekil 13. RESTful API ile öğretmenin profil bilgilerinin listelenmesi

Şekil 14. RESTful API ile randevu oluşturulması

Şekil 15. Veritabanında bulunan appointments tablosu

Şekil 16. Veritabanında bulunan mylessons tablosu

Şekil 17. RESTful API ile öğretmenin randevularının ve randevu isteklerinin listelenmesi

Şekil 18. RESTful API ile öğretmenin randevu durumunu güncellemesi

Şekil 19. Veritabanında bulunan appointments tablosu

Şekil 20. Veritabanında bulunan mylessons tablosu

Şekil 21. RESTful API ile öğrencilerin bütün öğretmenlere mesaj gönderememesi

Şekil 22. RESTful API ile öğrencinin öğretmene mesaj göndermesi

Şekil 23. RESTful API ile öğrencinin mesajlaştığı öğretmenlerin listelenmesi

Şekil 24. RESTful API ile mesajlaşmanın listelenmesi

Şekil 25. RESTful API ile öğrencinin aldığı derslerin listelenmesi

Şekil 26. RESTful API ile öğrencinin aldığı derse yorum yapması

Şekil 27. RESTful API ile eğitime yapılan yorumların listelenmesi

Şekil 28. Android Studio indirme

Şekil 29. Android Studio kurulum ilk adım

Şekil 30. Android Studio kurulum seçenekleri

Şekil 31. Android Studio kurulum izin seçimi

Şekil 32. Android Studio kurulumu yükleme

Şekil 33. Android Studio son kurulum adımı

Şekil 34. Android Studio kurulum tamamlama adımı

Şekil 35. Android hoş geldin sayfası

Şekil 36. Android Studio giriş sayfası

Şekil 37. Android Studio template seçimi

Şekil 38. Android Studio aktivite oluşturma adımı

Şekil 39. Android Studio örnek bir proje

Şekil 40. Volley kullanım örneği

Şekil 41. Mobil uygulama ana ekranı

Şekil 42. Mobil uygulama giriş ekranı

Şekil 43. Mobil uygulama kayıt seçme ekranı

Şekil 44. Mobil uygulama öğrenci kaydı

Şekil 45. Mobil uygulama eğitmen kaydı

Şekil 46. Mobil uygulama öğrenci ana sayfası

Şekil 47. Mobil uygulama öğrenci profili

- Şekil 48. Mobil uygulama öğrenci ders talebi
- Şekil 49. Mobil uygulama öğrenci kriterlerine uygun öğretmenler
- Şekil 50. Mobil uygulama öğretmen detay
- Şekil 51. Mobil uygulama alınan dersler
- Şekil 52. Mobil uygulama öğretmen mesajlaşma
- Şekil 53. Mobil uygulama öğretmen mesaj listesi
- Şekil 54. Mobil uygulama öğrenci mesaj gönderme
- Şekil 55. Mobil uygulama öğretmen değerlendirme
- Şekil 56. Mobil uygulama öğretmen ana sayfa
- Şekil 57. Mobil uygulama öğretmen profil
- Şekil 58. Mobil uygulama öğretmen mesaj listesi
- Şekil 59. Mobil uygulama öğretmen randevu onay/red
- Şekil 60. Mobil uygulama öğretmen mesajlaşma
- Şekil 61. Mobil uygulama öğretmen mesaj listesi
- Şekil 62. Mobil uygulama öğretmen mesaj gönderme

BÖLÜM 1

GİRİŞ

1.1. LİTERATÜR ÖZETİ

Eğitim bize etrafımızdaki dünya hakkında bilgi verir. İçimizde yaşama bakma perspektifini geliştirir. Fikirler oluşturmamıza ve yaşam üzerinde bakış açısına sahip olmamıza yardımcı olur. Eğitimciler bu süreçte en büyük destekçilerimizdir. Her öğrencinin kendine göre öğrenme düzeyi ve öğrenme için ayırması gereken süre farklılık gösterir. Son zamanlarda bu eksiği kapatmak için insanlar okula ilave olarak ek ders ihtiyacı duymaktadırlar.

Günümüzdeki en popüler şirketlerin verdiği hizmetlere baktığımızda seyahat hizmeti veren UBER gibi şirketlerin bir aracı olmadığını, konaklama hizmeti veren AIRBNB gibi şirketlerin bir otel ya da evlerinin olmadığını görüyoruz. Projemizdeki özel ders uygulamasında da sermayeye, binaya ya da dersane ortamına ihtiyaç yoktur. Eğitimde öğrenci de çalışma ortamlarını ve ulaşımını kendi imkanlarıyla sağlayabilirler.

1.2. TEZİN AMACI

Gelişen teknoloji ile bilgiye ulaşmak artık çok kolay ve ucuz. Youtube’ da yüzlerce eğitim videolarına tek tıkla ulaşabiliyoruz. App Store veya Google Play gibi online mağazalardan ilgili uygulamaları indirip online eğitim alabiliyoruz. Online eğitimdeki talep artmaya devam etse de herkes evinde uygun çalışma ortamına ya da gerekli disipline sahip olmayabilir.

Okul derslerinden aldığı eğitimi pekiştirmek isteyen öğrencilerin özel ders arayışı içinde olmaları fakat bunun için dershanelerin çok pahalı olması ve sadece istediği ders/konudan ders alma isteği insanları özel derse yönlendirmektedir.

Online mağazalardaki özel ders uygulamalarına baktığımızda, genel olarak uygulamalarda; çalışılacak ortamın seçimi, ücret, tarih vb. detaylar öğrenci ve öğretmen arasında konuşularak anlaşma sağlanmaktadır. Ücrette anlaşmazlık, derse gecikme, çalışma ortamı seçimindeki kararsızlık vb. sebepler taraflar arasında soruna yol açabilmekte ve kişiler bu durumdan mağdur olabilmektedir. Öğitmen profillerinin altında bazı yorumlar bulunmakta fakat bu yorumlar öğretmen hakkında yeterli bilgi vermemektedir. Bunun sebebi, ders alan her öğrencinin objektif yorum yapmaması ve ya öğrencilerin ders aldıkları öğretmeni değerlendirmemesidir..

Tezimizde, her ders sonrasındaki puanlama özelliği ile öğrencinin seçeceği öğretmen hakkında daha fazla bilgi sahibi olmasını amaçlıyoruz. Bu sayede öğrenciler, o öğretmen hakkındaki görüşlerini değerlendirip öğretmenin, o eğitim için uygun olup olmadığı hakkında fikir sahibi olabileceklerdir.

BÖLÜM 2

YÖNTEMLER, ARAÇLAR ve YAZILIMLAR

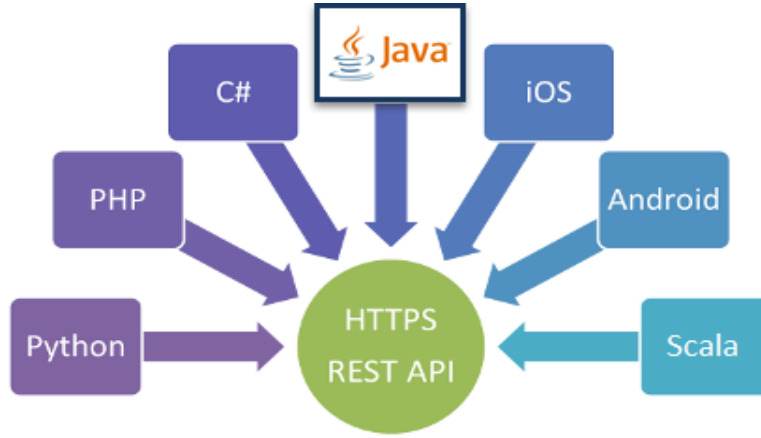
Proje RESTful API ve mobil uygulama olmak üzere iki kısımdan oluşmaktadır. RESTful API , esnek bir yapıya sahip, istemci – sunucu arasındaki basit bir veri iletişim yoludur. RESTful API, PhpStorm editörü kullanılarak Php programlama dili ile yazılmıştır. Programlama dilinden bağımsız olarak farklı platformlar arasında veri alışverişi sağladığından ve kolay yapısından esinlenerek veri alışverişinde JSON kullanılmıştır. Veritabanı teknolojisi için MYSQL tercih edilmiştir. Restful Apiler için test aşamasında, Postman teknolojisi kullanılmıştır. Mobil uygulama, Android Studio platformunda Java programlama dili ile geliştirilmiştir.

2.1. REST TEKNOLOJİSİ

REST, Representational State Transfer ‘in kısaltmasıdır. Representational State Transfer yani Temsili Durum Transferi, HTTP metotlarını kullanan, esnek bir yapıya sahip, istemci – sunucu arasındaki basit bir veri iletişim yoludur. Katı kurallara sahip bir protokolden ziyade bir mimari stildir.

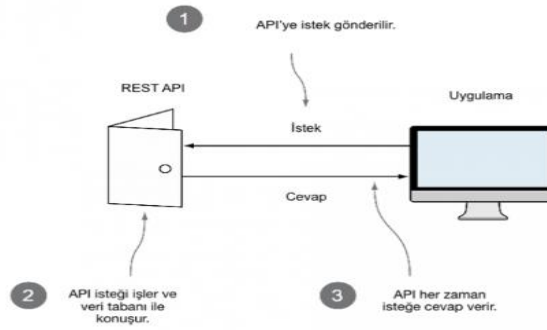
REST mimarisi kullanılan dil ve format bakımından gayet esnektir. JSON, XML hatta TEXT dosyasıyla bile veri alışverişi yapabilmemizi sağlar. JSON ile aynı veriyi çok daha düşük boyutla taşıyabilir. Web serviste çalışacak metot URI (Uniform Resource Identifier) yani Tekdüzen Kaynak Tanımlayıcısı ile HTTP üzerinden çağırılır [1].

API(Application Programming Interface), herhangi bir uygulamanın belli işlevlerini diğer uygulamalarında kullanabilmesi için oluşturulmuş bi modüldür. Uygulamaların birbirleriyle konuşmasını sağlar.



Şekil 1. REST API'nin bütün programlama dilleri ile çalışması

REST API programlama dili farketmeksizin, gelen isteklere cevaplar döndürür. Tercih edilmesindeki en büyük sebeplerden biride budur.



Şekil 2. REST API çalışma şekli

Kısaca özetleyecek olursak, Şekil 2’de de görüldüğü üzere uygulama istekte bulunur. REST API’de isteği işler ve veri tabanında gerekli işlemleri yapar. İstenilen formatta cevapları uygulamaya döndürür.

REST ile yazılmış bir servisle çalışmak için ihtiyacımız olan tek şey URL’dir. Bir URL’i çağırdığımızda URL, JSON veya XML döndürür, dönen cevabı parse ederek, servis entegrasyonu tamamlanmaktadır. Yani teorik olarak istemci uygulama REST bir servisin yapısını ve detaylarını bilmek zorunda değildir. REST’in son derece esnek bir yapısı vardır.

Http, web tarayıcısı ile web sunucu arasında iletişim kurmak için kullanılan protokoldür. Açılımı Hyper Text Transfer Protocol şeklindedir. Http’de metodlar kullanılarak sunucuya istek gönderilir ve gönderilen istekler karşılığında sunucudan cevaplar alınır. HTTP istekleri sunucuya hangi tür işlem yapılacağını söyleyen farklı metotlara sahiptir

Başlıca Http metodları: Get, Post, Put, Delete, Connect, Options, Trace, Patch dir. Bu metodlardan sıkça kullanılan Http metodları Get ve Post’tur [2].

GET Metodu: Bu metod sunucudan veri almak için kullanılır.

POST Metodu: Bu metod ile sunucuya veri yazdırmak için kullanılır..

PUT Metodu: Bu metod ile servis sağlayıcı üzerindeki bir kaynak güncellenebilir.

DELETE Metodu: Bu metod ile sunucudaki herhangi bir veri silinebilir.

2.1.2. REST Mimari Özellikleri

REST mimarî tarzıyla etkilenen mimarî özellikler şunlardır [3]:

- Performans - bileşen etkileşimleri, kullanıcının tespit ettiği performans ve ağ verimliliğinde dominant faktör olabilir.
- Çok sayıda bileşeni ve bileşenler arası etkileşimi ölçeklenebilirlik yapmak.
- Arayüzlerin sadeliği,
- Değişen ihtiyaçları karşılamak için (uygulama yürütülürken dahi) bileşenlerin değiştirilebilirliği,
- Hizmet vekillerince (İng. *İngilizce: service agents*) tarafından bileşenlerarası haberleşmenin görünürlüğü
- Program kodunu veriyle beraber hareket ettirerek bileşenlerin taşınabilirliği,
- Sistem seviyesinde bileşenlerde, bağlantılarda veya verideki hatalara rağmen hataya karşı direnç güvenilirliği.
- Bir servisin restful olması için temel karakteristik prensipler vardır:
 - Client-Server yapısı: Client, server tarafındaki veri kaynağı ile ilgili hiç birşey bilmediği gibi, server da client hakkında birşey bilmemelidir. Server sadece kendisine gelen istekler doğrultusunda değer döndürür. Böylece server daha basit ve ölçeklenebilir olur.
 - Stateless: Server tarafında client ile ilgili hiç bir bilginin tutulmamasıdır. Client tarafından gerçekleştirilecek olan her request, serverin response dönebilmesi için geçerli bilgiyi taşır. Client ayrımı ise clientin servera request geçme esnasında göndereceği bir token veya kimlik bilgisi ile gerçekleştirilir.
 - Cacheable: Tüm HTTP response lar client tarafından cache'lenebilir olmalı. Veri tutarlılığını sağlayabilmek için ise server ilgili response bilgisini dönerken, ilgili verinin cache'lenebilir olup olmadığını göndermesi gerekir.

- Layered System: Clientın server yapısındaki hangi katmana bağlandığını bilmemesidir. Bu bir ara katmanda olabilir, son katmanda. Buradaki amaç ise serverlar arasında load balancing yaparak ölçeklenebilirliği artırmak ve clientları bazı güvenlik politikalarına da zorlayabilmektedir.
- Uniform Interface: Client ile server arasındaki ortak bir arayüzün olması ile birbirlerinden bağımsız bir şekilde geliştirilebiliyor olurlar. Bir servisin RESTful olabilmesi için en önemli prensiplerden birisidir.

2.2. RESTFUL API GELİŞTİRİLİRKEN KULLANILAN TEKNOLOJİLER

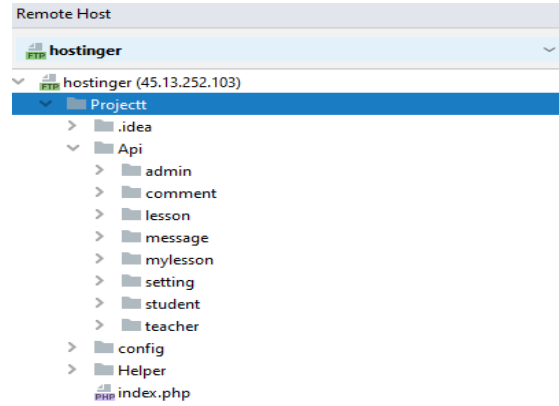
RESTful API geliştirilirken PhpStorm teknolojisi ve veritabanı için Mysql kullanılmıştır. Veri iletişimi için JSON veri yapısı kullanılmıştır.

Oluşturmuş olduğumuz RESTful API dosyalarına farklı sunuculardan erişebilmek için WebHostinger sitesinden host ve heryerdeegitim.space adlı domain alınmıştır. Oluşturmuş olduğumuz ilişkili veritabanı ve RESTFUL API dosyaları hostingere yüklenmiştir.

Projemizde www.heryerdeegitim.space URL'i kullanılarak HTTP metotları ile gerekli işlemler yapılmaktadır.

2.2.1.PHPStorm

Editör olarak PhpStorm kullanılmıştır. Şekil 3'te projemizin RESTful API mimarisi gösterilmektedir. Öncelikle gelen istekler index.php sayfasından yönlendirilmektedir (Şekil 4).



Şekil 3. RESTful API dizin yapısı

```
<?php
header( string: 'Access-Control-Allow-Origin: *');
header( string: "Content-Type:application/json;charset=utf8");
require_once 'config/database.php';
require_once 'Helper/mHelper.php';
$db=new Database();
$returnArray=[];

$mode=$_GET['mode'];
$process=$_GET['process'];

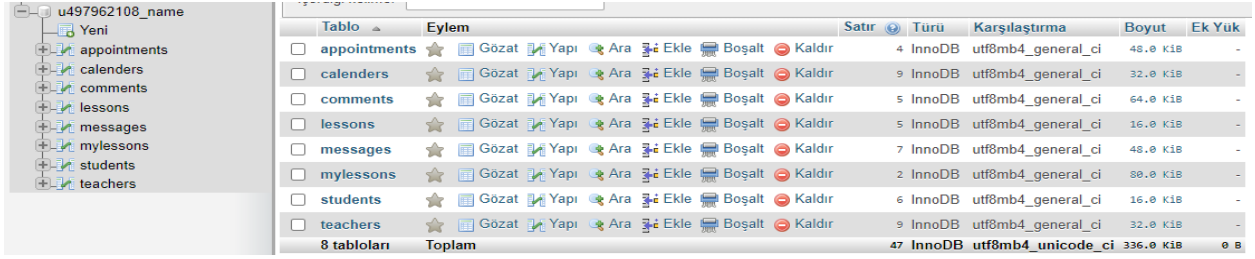
$path='Api/'.$mode.'/'.$process.'.php';

if(file_exists($path)){
    require_once('Api/'.$mode.'/'.$process.'.php');
    echo json_encode($returnArray);
}
else{
    die("Page is Not Found");
}
?>
```

Şekil 4. RESTful API index.php

2.2.2.MySql-PhpMyAdmin

PhpMyAdmin,php ile yazılmış,açık kaynak kodlu, web tabanlı bir veritabanı yönetim yazılımıdır. Başlıca kullanım alanı internet üzerinden MySQL veritabanı yönetimidir. Veritabanı oluşturma ve silme, tablo ekleme/değiştirme/silme, alan ekleme/değiştirme/silme, SQL sorguları çalıştırma, kullanıcıları, yetkileri ve alan anahtarlarını yönetme gibi işlevleri pratik ve hızlı bir şekilde yapabilen ücretsiz bir yazılım olduğundan tezimizde PhpMyAdmin tercih edilmiştir [4].



Tablo	Eylem	Satır	Turu	Karşılaştırma	Boyut	Ek Yük
<input type="checkbox"/> appointments	Gözet Yapı Ara Ekle Boşalt Kaldır	4	InnoDB	utf8mb4_general_ci	48.0 KİB	-
<input type="checkbox"/> calenders	Gözet Yapı Ara Ekle Boşalt Kaldır	9	InnoDB	utf8mb4_general_ci	32.0 KİB	-
<input type="checkbox"/> comments	Gözet Yapı Ara Ekle Boşalt Kaldır	5	InnoDB	utf8mb4_general_ci	64.0 KİB	-
<input type="checkbox"/> lessons	Gözet Yapı Ara Ekle Boşalt Kaldır	5	InnoDB	utf8mb4_general_ci	16.0 KİB	-
<input type="checkbox"/> messages	Gözet Yapı Ara Ekle Boşalt Kaldır	7	InnoDB	utf8mb4_general_ci	48.0 KİB	-
<input type="checkbox"/> mylessons	Gözet Yapı Ara Ekle Boşalt Kaldır	2	InnoDB	utf8mb4_general_ci	88.0 KİB	-
<input type="checkbox"/> students	Gözet Yapı Ara Ekle Boşalt Kaldır	6	InnoDB	utf8mb4_general_ci	16.0 KİB	-
<input type="checkbox"/> teachers	Gözet Yapı Ara Ekle Boşalt Kaldır	9	InnoDB	utf8mb4_general_ci	32.0 KİB	-
8 tabloları	Toplam	47	InnoDB	utf8mb4_unicode_ci	336.0 KİB	0.8

Şekil 5. Veritabanı tabloları

Şekil 5’te bulunan veritabanı tabloları, WorkBench kullanılarak ilişkili bir şekilde oluşturulmuştur ve hostingere yüklenmiştir.

```
<?php
class Database
{
    public $db;

    function __construct()
    {
        try {
            $this->db = new PDO( dsnc: 'mysql:host=45.13.252.149;dbname=u497962108_name;charset=utf8', username: "u497962108_user", passwd: "123456");
        }
        catch (PDOException $e)
        {
            echo $e->getMessage();
        }
    }
}
```

Şekil 6. RESTful API ile veritabanı bağlantısının yapılması

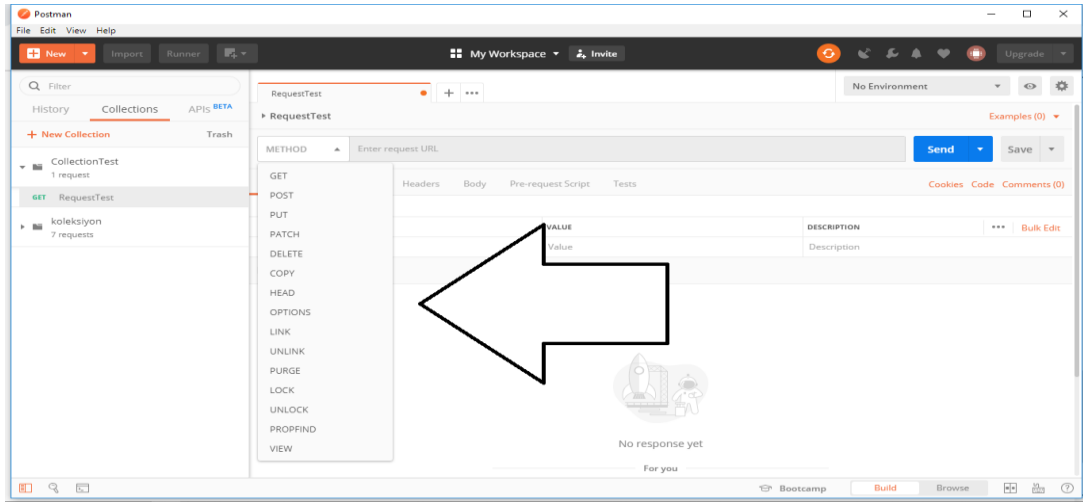
Veritabanı bağlantısı RESTful API'den şekildeki gibi yapılarak, RESTful API'nin veritabanına erişimi sağlanmıştır.

2.2.3.Postman

Postman, 3 milyondan fazla yazılımcının API'ları paylaşmak, test etmek, dokümente etmek, monitör etmek için kullandığı, dünyanın en çok kullanılan "rest-client". En öne çıkan özelliği tüm bunlar için çok kullanışlı bir arayüz sunmasıdır.

API, farklı uygulama yazılımlarının birbiriyle iletişim halinde olmasını sağlayan "Uygulama Programlama Arayüzü" anlamına gelir. Bir geliştirici, bir uygulama yazılımı oluştururken veya geliştirirken API'yı kullandığında zaman ve enerji tasarrufu sağlayabilir. Aynı zamanda, API kullanılarak kaynaklara kontrol erişimi ve hizmetler arasındaki iletişime erişim sağlanabilir.

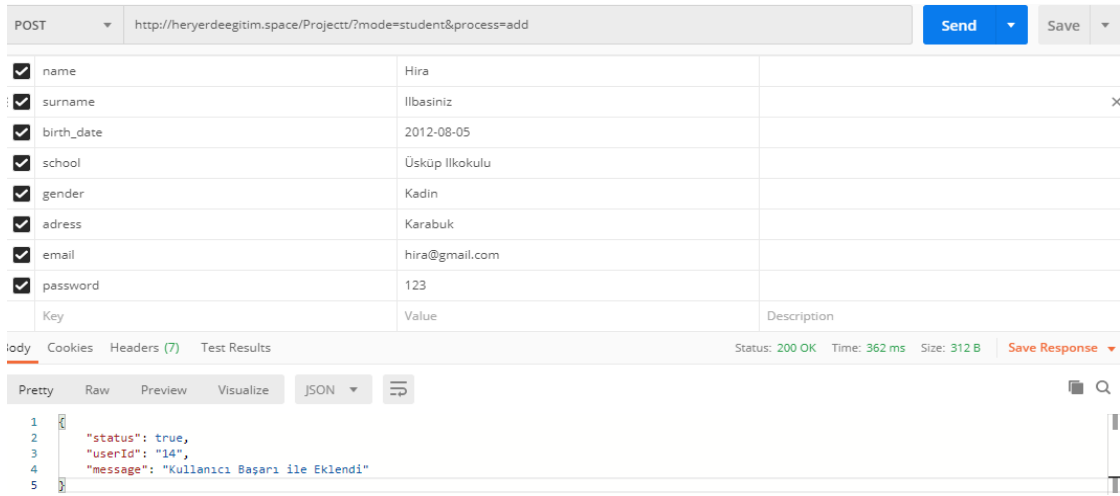
Postman, uzun kodlara ihtiyaç duymadan API'larınızı kolayca test edebileceğiniz, API'nızın dokümasını hazırlamanızı sağlayan ve "rest client" olarak adlandırabileceğiniz bir araçtır. Postman ayrıca Chrome için bir eklenti olarak da kullanılabilir [5].



Şekil 7. Postman arayüzü ve Http metodları

2.3.RESTFUL API'nin TEST EDİLMESİ

Projemizde Postman teknolojisi kullanılarak RESTful API'ler test edilmiştir. Postman aracılığı ile RESTful API'ye istek gönderilmekte ve isteklere cevap olarak json formatta veriler elde edilmektedir.



Şekil 8. RESTful API ile öğrenci kayıt

Öğrenci eklenirken Post metodu kullanılarak gerekli bilgiler RESTful API aracılığı ile veri tabanına kaydedilir (Şekil 8).

Öğrenci ve eğitmene ait kayıt işleminde alınan bilgiler sırasıyla students ve teachers tablolarında bulunmaktadır.

POST http://heryerdeegitim.space/Projectt/?mode=teacher&process=add

name	Helin
surname	Aydin
birth_date	1990-05-03
school	AÜ
gender	Kadin
adress	Ankara
email	helin@gmail.com
password	123
lessons_id	9
price	170
Key	Value

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "teachers_id": "37",
4   "calenders_id": "21",
5   "message": "Kullanıcı Başarı ile Eklendi"
}

```

Şekil 9. RESTful API ile eğitimci kayıt

Eğitimci için kayıt alınırken verdiği dersin id'si (lessons_id) ile birlikte kayıt alınmaktadır (Şekil9).

Eğitimcilerin boş gün bilgisini girebilmesi için her bir eğitimci için calenders tablosunda ayrı bir alan ayrılmaktadır. Eğitimci kayıt olduktan sonra uygun oldukları tarih ve saat bilgisini buradan istedikleri zaman güncelleyebilecektir.

SELECT * FROM `calenders`

☐ Tümünü göster | Satır sayısı: 25 | Satırları süz: Bu tabloda ara

+ Seçenekler

	id	hour	time	date	teachers_id
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	4	2	17:00:00	2020-05-31	14
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	5	2	15:00:00	2020-05-30	15
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	7	1	15:00:00	2020-05-29	17
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	11	2	17:00:00	2020-06-02	22
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	12	2	18:00:00	2020-06-10	23
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	13	1	12:00:00	2020-06-06	24
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	14	2	15:00:00	2020-05-29	25
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	15	1	16:00:00	2020-05-25	26
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	21	NULL	NULL	0000-00-00	37

Şekil 10. Veritabanında bulunan calenders tablosu

Uygulamaya kaydolun eğitime calenders tablosunda, boş gün bilgisini güncelleyebileceği bir alan oluşturulmuştur. Hour(dersin kaç saat verileceği), time ve date alanları ilk başta null olarak atanmıştır. (Şekil 10)



Şekil 11. RESTful API ile eğitmenin boş gün bilgisini güncellemesi

Eğitmenlerin boş gün bilgisi; kendisine ait calenders tablosundaki id'si ve eğitmenin id'si(teachers_id) ile birlikte Şekil 11'deki gibi POST metodu ile güncellenmektedir.

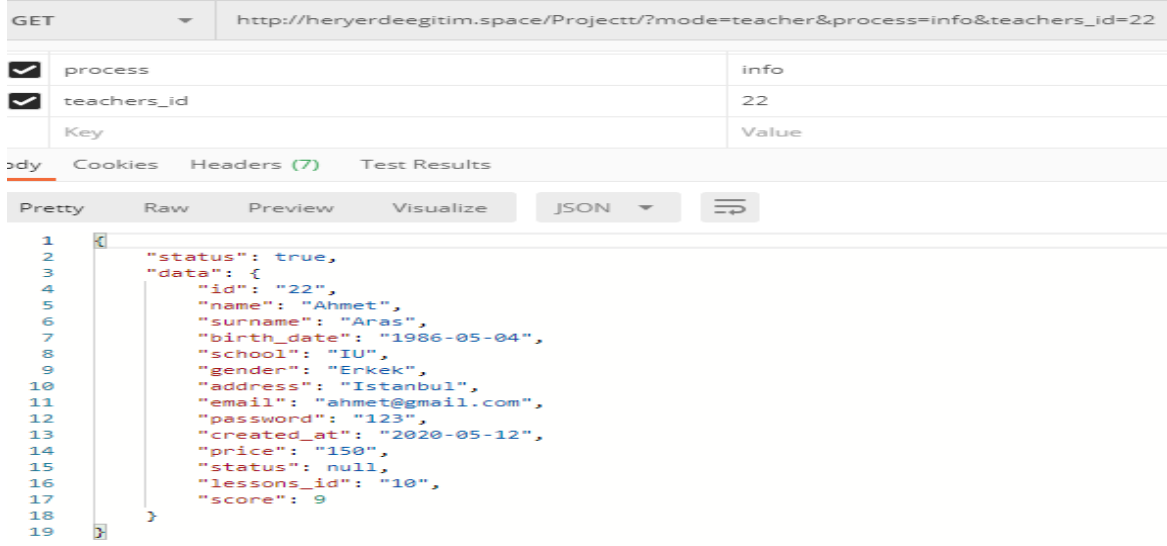
```
GET http://heryerdeegitim.space/Projectt/?mode=student&process=lessonsearch&address=Istanbul&lessons_id=10

Pretty Raw Preview Visualize JSON

{
  "id": "22",
  "teacher": "Ahmet Aras",
  "address": "Istanbul",
  "price": "150",
  "date": "2020-06-02",
  "time": "17:00:00",
  "hour": "2"
},
{
  "id": "23",
  "teacher": "Mustafa Yılmaz",
  "address": "Istanbul",
  "price": "200",
  "date": "2020-06-10",
  "time": "18:00:00",
  "hour": "2"
},
{
  "id": "25",
  "teacher": "Ali Kara",
  "address": "Istanbul",
  "price": "150",
  "date": "2020-05-29",
  "time": "15:00:00",
  "hour": "2"
},
--
```

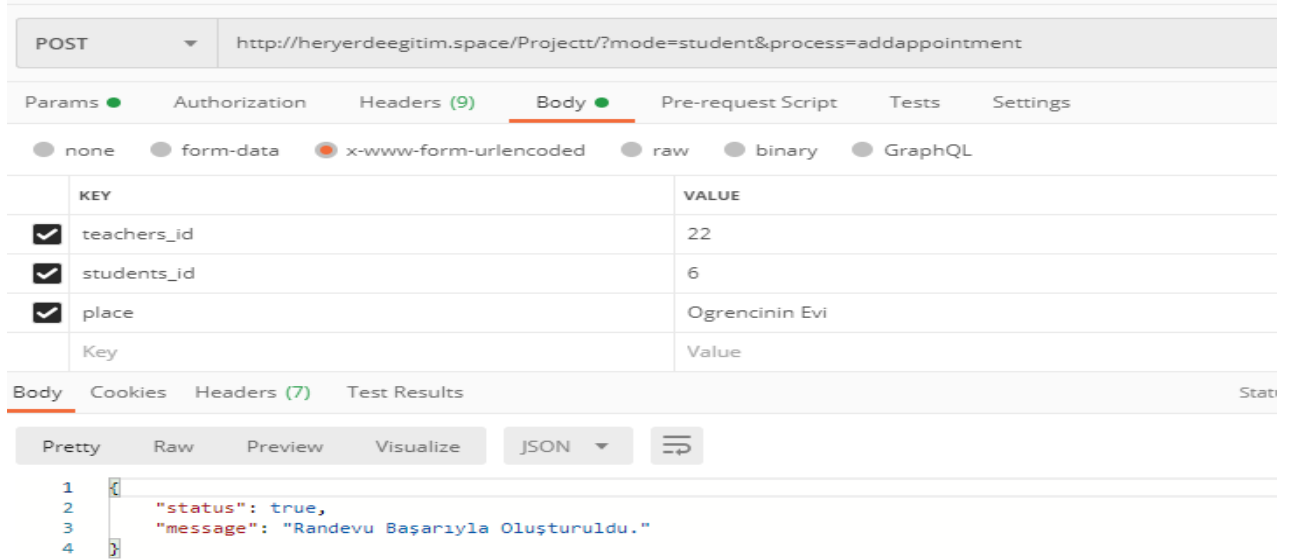
Şekil 12 . RESTful API ile öğrencinin ders araması

Öğrenciler kendilerine uygun ders araması yaparken filtre olarak istedikleri dersi(lessons_id) ve adres (address) bilgisini girmektedir. GET metodu ile birlikte istenen filtrelere uygun eğitimciler ve onlara ait; ücret,tarih, saat ve dersin kaç saat verileceği gibi bilgiler listelenmektedir (Şekil 12).



Şekil 13. RESTful API ile eğitmenin profil bilgilerinin listelenmesi

Öğrencilerin incelemek istedikleri eğitmenler, eğitmenin id'si(teachers_id) ile birlikte GET metodu kullanılarak listelenmektedir (Şekil 13).



Şekil 14. RESTful API ile randevu oluşturulması

Öğrencinin ders arama sonucunda; kendisine uygun bulduğu eğitmeni seçmiş olmasıyla, eğitmenin id'si(teachers_id) ve dersi almak istediği yer bilgisi(place) POST edilerek randevu oluşturulur (Şekil 14).

SELECT * FROM `appointments`

☐ Tümüü göster | Satır sayısı: 25 | Satırları süz: Bu tabloda ara

Seçenekler

	id	created_at	place	status	students_id	teachers_id
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	37	2020-05-19	Öğrencinin Evi	Beklemede	6	22

Şekil 15. Veritabanında bulunan appointments tablosu

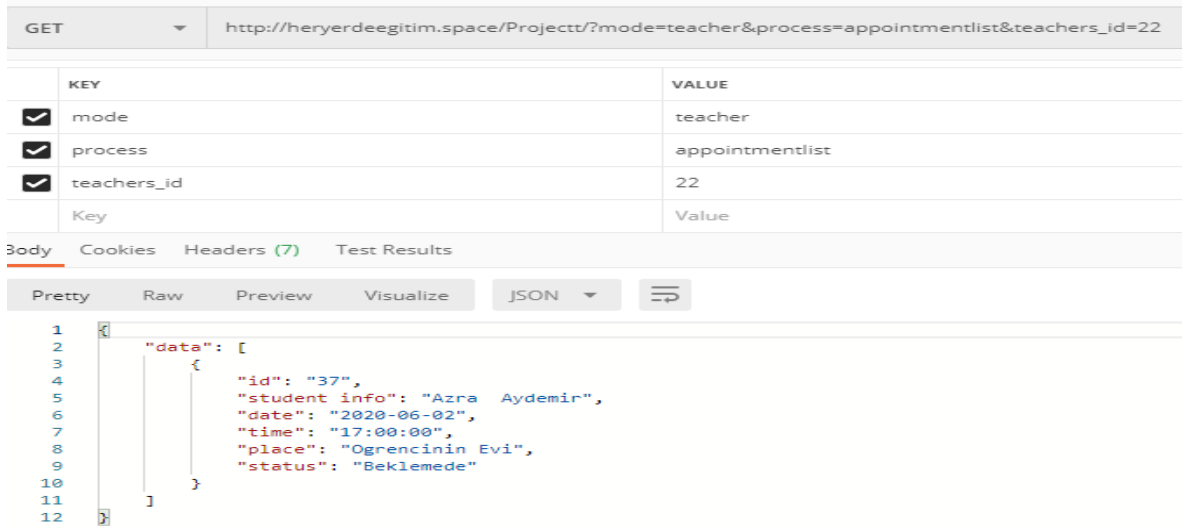
Randevu oluşturulduğunda, randevunun durum bilgisi Şekil 15'te görüldüğü gibi “Beklemede” şeklinde veritabanında ki appointments adlı tabloda yer almaktadır. Ayrıca randevu oluşturulduğunda appointments tablosuyla birlikte öğrencinin almış olduğu randevuya ait bilgiler mylessions tablosuna da POST edilir.

	id	lessonname	teachers	hour	price	place	date	time	status	lessons_id	teachers_id	students_id	appointments_id
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	16	Fen Bilgisi	Ahmet Aras	2	150	Öğrencinin Evi	2020-06-02	17:00:00	Beklemede	10	22	6	37

Seçilileri: ☐ Düzenle ☐ Kopyala ☐ Sil ☐ Dışa aktar

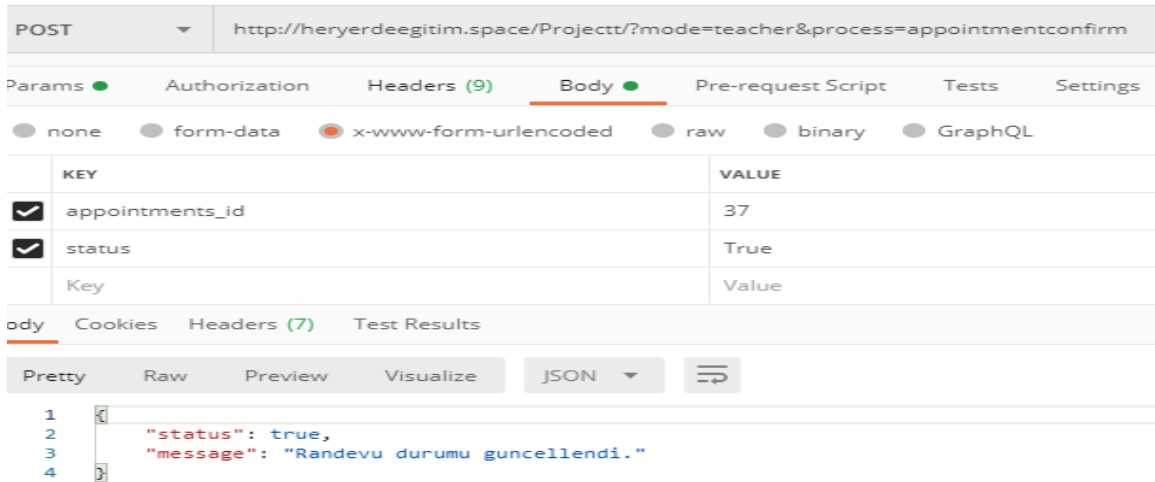
Şekil 16. Veritabanında bulunan mylessions tablosu

Şekil 16'da mylessions tablosunda tutulan veriler gösterilmektedir. Öğrenci bu tablo sayesinde veritabanından çekilen veriler ile almış olduğu ders bilgilerini ve randevu bilgilerini takip edebilmektedir.



Şekil 17. RESTful API ile eğitmenin randevularının ve randevu isteklerinin listelenmesi

Eğitmenler gelen randevu isteklerini ve randevularını kendi id'leri ve GET metoduyla Şekil 17'deki gibi listeleyebilmektedirler. JSON formatında o randevuya ait id, randevu isteğinde bulunan öğrencinin bilgisi, tarihle ilgili detaylar, randevu yeri ve durumu listelenmektedir.






Şekil 18 . RESTful API ile eğitmenin randevu durumunu güncellemesi

Eğitmen seçmiş olduğu randevunun durum bilgisini “True” yaparak onaylayabilir yada “False” yaparak reddedebilir (Şekil 18).

				id	created_at	place	status	students_id	teachers_id
				37	2020-05-19	Oğrencinin Evi	True	6	22

Şekil 19. Veritabanında bulunan appointments tablosu

				id	lessonname	teachers	hour	price	place	date	time	status	lessons_id	teachers_id	students_id	appointments_id
				16	Fen Bilgisi	Ahmet Aras	2	150	Oğrencinin Evi	2020-06-02	17:00:00	True	10	22	6	37

Şekil 20. Veritabanında bulunan mylessons tablosu

Öğretmen durum bilgisini istediği şekilde POST ettikten sonra, appointments ve mylessons tablolarının da durum bilgisi güncellenir (Şekil 19 ve Şekil 20).

POST

http://heryerdeegitim.space/Projectt/?mode=student&process=sendMessage

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

KEY

VALUE

students_id

6

teachers_id

22

text

Merhaba Hocam

Key

Value

body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

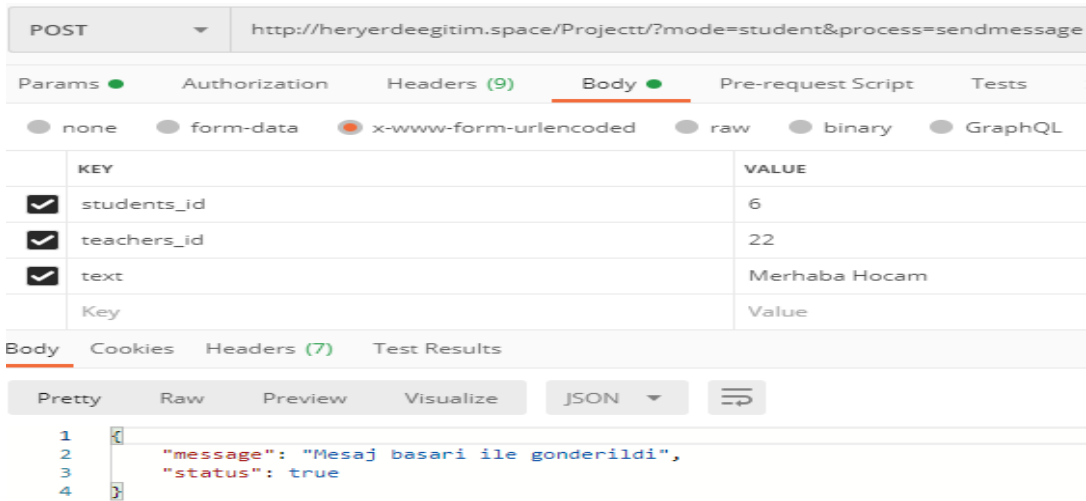
3

"message": "Bu egitimene mesaj gonderemezzsiniz."

Şekil 21 . RESTful API ile öğrencilerin bütün eğitmenlere mesaj gönderememesi

Öğrenci ve öğretmen arasında mesajlaşma işlemi ancak randevu onaylandığı zaman gerçekleşebilir. Veritabanında messages adlı veritabanında ilgili veriler tutulmaktadır.

Eğer randevu öğretmen tarafından onaylanmamışsa veya öğrenci o öğretmeninden daha öncede ders almadıysa, öğrenci veya öğretmen birbirine mesaj gönderememektedir (Şekil 21).



Şekil 22. RESTful API ile öğrencinin öğretmene mesaj göndermesi

Randevu durumu onaylanmış ise, öğrencinin id'si (students_id), öğretmenin id'si(teachers_id) ve mesaj(text) POST edilerek mesaj başarılı bir şekilde gönderilebilmektedir. (Şekil 22)

GET http://heryerdeegitim.space/Projectt/?mode=student&process=selectmessage&students_id=6

KEY	VALUE
mode	student
process	selectmessage
students_id	6

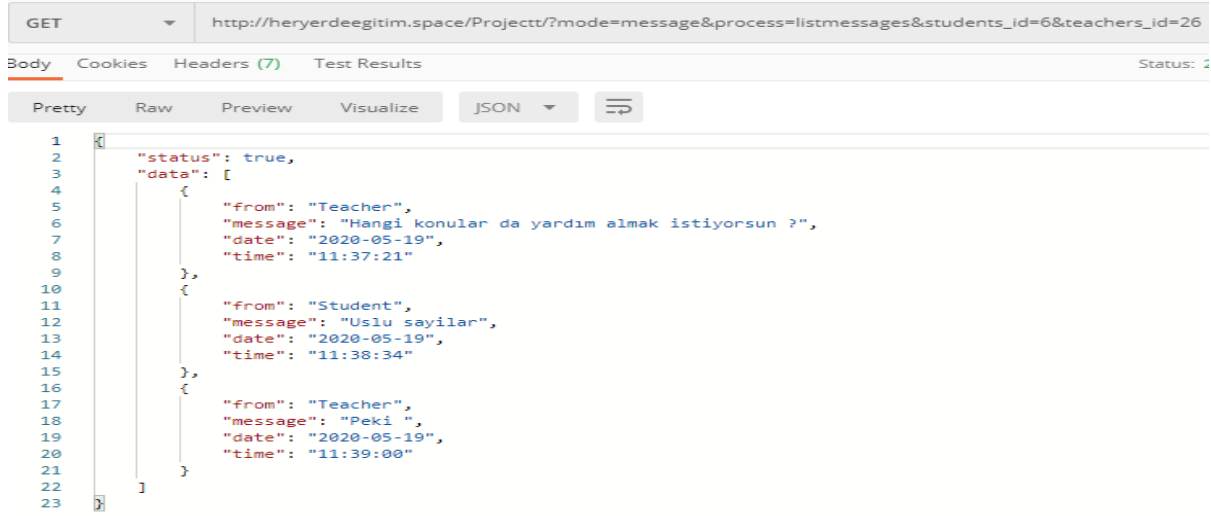
body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "data": [
4     {
5       "teachers_id": "22",
6       "teacher": "Ahmet Aras"
7     },
8     {
9       "teachers_id": "26",
10      "teacher": "Arda Güney"
11    }
12  ]
13 }
```

Şekil 23. RESTful API ile öğrencinin mesajlaştığı öğretmenlerin listelenmesi

Öğrenci veya öğretmen uygulamada mesajlarım kısmını görüntülerken kendi id'si ve GET metoduyla listeleyebilecektir. Şekilde json olarak gelen verilerde öğrencinin mesajlaştığı öğretmenlerin id'si ve ad soyad bilgisi gelmektedir. (Şekil 23)



```
GET http://heryerdeegitim.space/Projectt/?mode=message&process=listmessages&students_id=6&teachers_id=26

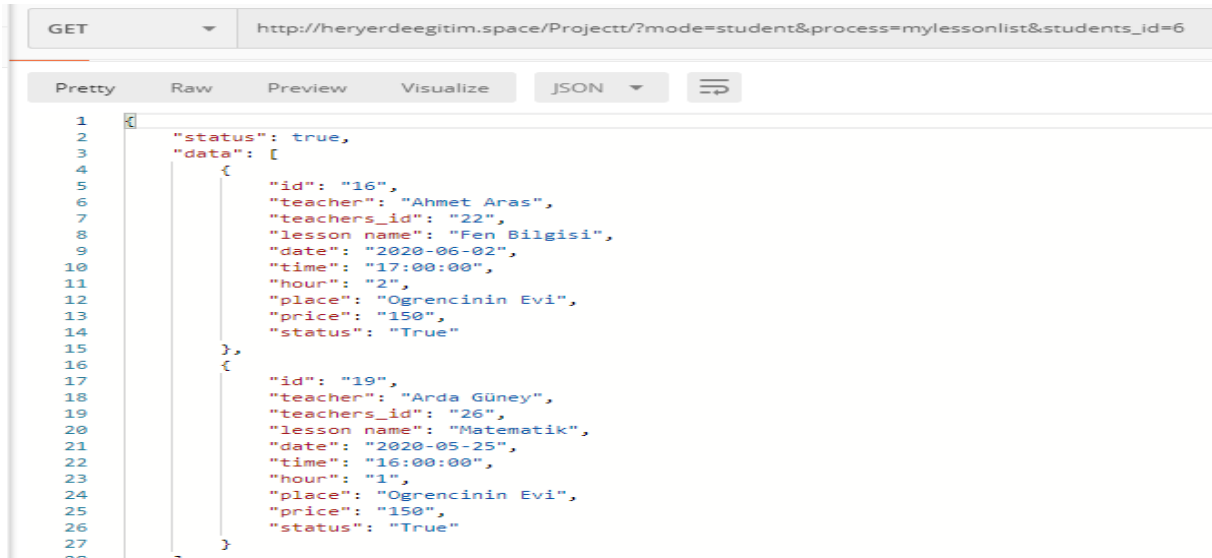
Body Cookies Headers (7) Test Results Status: 200

Pretty Raw Preview Visualize JSON

1 {
2   "status": true,
3   "data": [
4     {
5       "from": "Teacher",
6       "message": "Hangi konular da yardım almak istiyorsun ?",
7       "date": "2020-05-19",
8       "time": "11:37:21"
9     },
10    {
11      "from": "Student",
12      "message": "Uslu sayilar",
13      "date": "2020-05-19",
14      "time": "11:38:34"
15    },
16    {
17      "from": "Teacher",
18      "message": "Peki ",
19      "date": "2020-05-19",
20      "time": "11:39:00"
21    }
22  ]
23 }
```

Şekil 24. RESTful API ile mesajlaşmanın listelenmesi

Öğrencinin seçmiş olduğu öğretmenin id'si ile birlikte, aralarında geçen mesajlaşma bu şekilde listelenmektedir (Şekil 24).



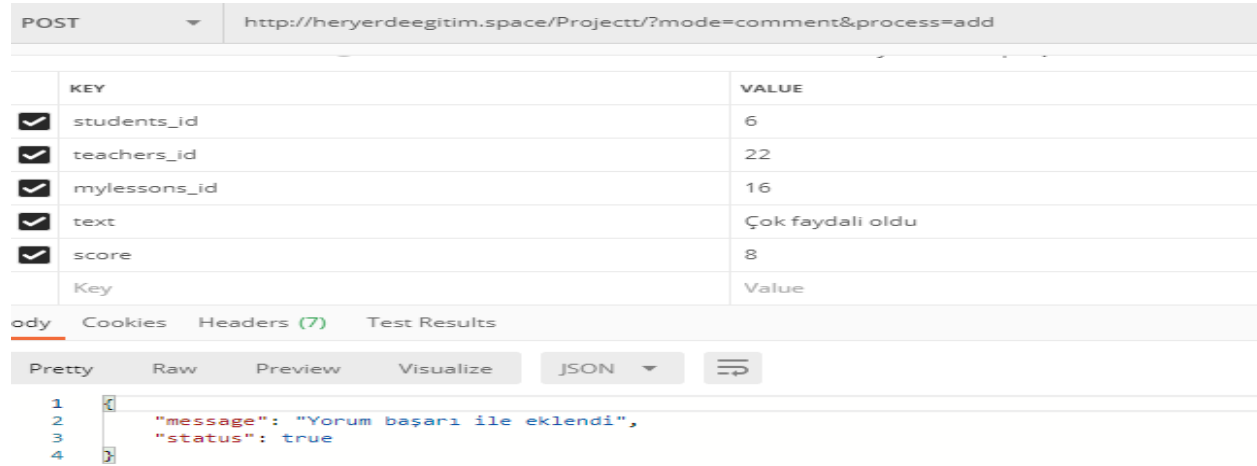
```
GET http://heryerdeegitim.space/Projectt/?mode=student&process=mylessonlist&students_id=6

Pretty Raw Preview Visualize JSON

1 {
2   "status": true,
3   "data": [
4     {
5       "id": "16",
6       "teacher": "Ahmet Aras",
7       "teachers_id": "22",
8       "lesson name": "Fen Bilgisi",
9       "date": "2020-06-02",
10      "time": "17:00:00",
11      "hour": "2",
12      "place": "Oğrencinin Evi",
13      "price": "150",
14      "status": "True"
15    },
16    {
17      "id": "19",
18      "teacher": "Arda Güney",
19      "teachers_id": "26",
20      "lesson name": "Matematik",
21      "date": "2020-05-25",
22      "time": "16:00:00",
23      "hour": "1",
24      "place": "Oğrencinin Evi",
25      "price": "150",
26      "status": "True"
27    }
28  ]
29 }
```

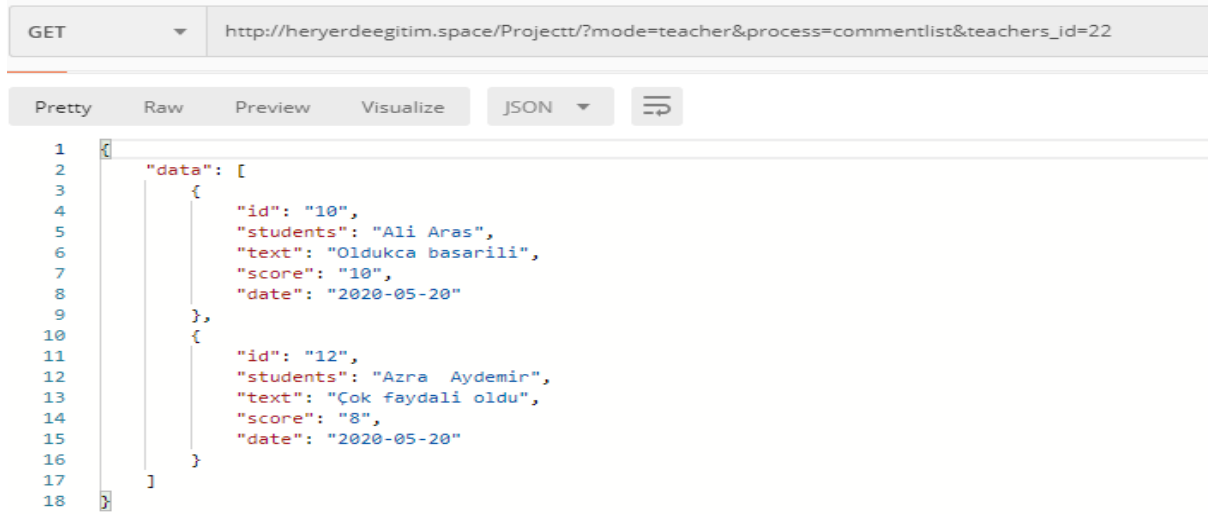
Şekil 25. RESTful API ile öğrencinin aldığı derslerin listelenmesi

Öğrencilerin aldığı dersler, kendi id'si(students_id) ve GET methodu ile birlikte listelenmektedir (Şekil 25).



Şekil 26. RESTful API ile öğrencinin aldığı derse yorum yapması

Öğrenciler aldıkları derslerle ilgili olarak öğretmenlere yorum yapıp puan verebilmektedirler. İlgili öğrencinin id'si(students_id), öğretmenin id'si(teachers_id), alınan dersin id'si(mylessons_id), yorum(text) ve puan(score) verileri şekildeki gibi POST edilerek eğitmene yorum yapıp, puan verilebilmektedir. Yorum ve puanlar öğrencilerin ders seçimlerinde etkili olmaktadır (Şekil 26).



The screenshot shows a REST client interface. At the top, the method is 'GET' and the URL is 'http://heryerdeegitim.space/Projectt/?mode=teacher&process=commentlist&teachers_id=22'. Below the URL bar, there are tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize', with 'Pretty' selected. To the right of these tabs is a dropdown menu set to 'JSON' and a refresh icon. The main area displays the JSON response, which is formatted and includes line numbers on the left. The response is an array with two objects, each containing 'id', 'students', 'text', 'score', and 'date' fields.

```
1 {
2   "data": [
3     {
4       "id": "10",
5       "students": "Ali Aras",
6       "text": "Oldukca basarili",
7       "score": "10",
8       "date": "2020-05-20"
9     },
10    {
11      "id": "12",
12      "students": "Azra Aydemir",
13      "text": "Çok faydali oldu",
14      "score": "8",
15      "date": "2020-05-20"
16    }
17  ]
18 }
```

Şekil 27. RESTful API ile eğitmene yapılan yorumların listelenmesi

Ders arama sonuçları listelendiğinde öğrencinin istediği eğitmenin id'si(teachers_id) ve GET metodu ile birlikte, puan ve yorum bilgilerine ulaşp o eğitmen hakkında varsayımda bulunabilirler (Şekil 27).

2.4.ANDROID MOBİL UYGULAMA GELİŞTİRİLMESİ

Android; Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilmekte olan, Linux tabanlı özgür ve ücretsiz bir işletim sistemidir. Sistem açık kaynak kodlu olsa da, küçük ama önemli bir kısmı Google tarafından kapalı tutulmaktadır. Ücretsiz olmasının sebebi, sistemin daha hızlı ve çabuk gelişmesi, birçok popüler marka tarafından kullanılması ve bu sayede reklamlarının daha fazla kişiye ulaşmasını sağlamaktır. Google, Android üzerinde çalışan Google Play marketteki oyun ve uygulamalar üzerindeki reklamları yayınlayarak para kazanmaktadır. Android' in desteklenen uygulama uzantısı ".apk"dır.

Android, aygıtların fonksiyonelliğini genişleten uygulamalar yazan geniş bir geliştirici grubuna sahiptir. Android tabanlı 1 milyondan fazla uygulama bulunmaktadır.

Open Handset Alliance, 5 Kasım 2007'de Android'i kurduğunu duyurmuştur ve ardından 34 adet donanım, yazılım ve telekom şirketi, mobil cihazlar için telif hakkı olmayan bir işletim sisteminin teknolojinin gelişimi için yararlı olduğu konusunda hemfikir olmuşlardır. Open Handset Alliance, 5 Kasım 2007'de Android'i kurduğunu duyurmuştur ve ardından 34 adet donanım, yazılım ve telekom şirketi, mobil cihazlar için telif hakkı olmayan bir işletim sisteminin teknolojinin gelişimi için yararlı olduğu konusunda hemfikir olmuşlardır.

Android, Linux çekirdeği üzerine inşa edilmiş bir mobil işletim sistemidir. Bu sistem ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu Java-uyumlu kütüphaneleri içine alan uygulama iskeleti üzerinden çalışmaktadır. Android, derlenmiş Java kodunu çalıştırmak için dinamik çevirmeli Android Runtime (ART) kullanır ve cihazların fonksiyonelliğini artıran uygulamaların geliştirilmesi için çalışan geniş bir programcı-geliştirici çevresine sahiptir.

2.4.1 Android Mobil Uygulama Geliştirirken Kullanılan Teknolojiler

Mobil uygulama geliştirirken IDE (Integrated Development Environment) olarak Android Studio kullanılmıştır. Uygulamada ListView, Spinner, Volley ve Gson teknolojileri kullanılmıştır.,

2.4.2 Android Studio

Android Studio, Android için resmi tümleşik geliştirme ortamıdır. 16 Mayıs 2013 tarihinde Google I/O etkinliğinde tanıtılmıştır. Android Studio, IntelliJ IDEA'ya dayalı olup Android geliştirme için özel olarak tasarlanmıştır.

Android Studio, Apache lisansı ile lisanslanmıştır ve ücretsiz olarak indirilip kullanılabilir. Aşağıda Android Studio Windows için kurulumu ve örnek bir proje oluşturulması gösterilmiştir.

Öncelikle Android Studio'nun sitesinden indirme işlemini gerçekleştirelim.

Download Android Studio

Before downloading, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <https://source.android.com/>, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (<https://source.android.com/compatibility>) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).

1.4 "Google" means Google LLC, organized under the laws of the State of Delaware, USA, and operating under the laws of the USA with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA.

2. Accepting this License Agreement

☐ I have read and agree with the above terms and conditions

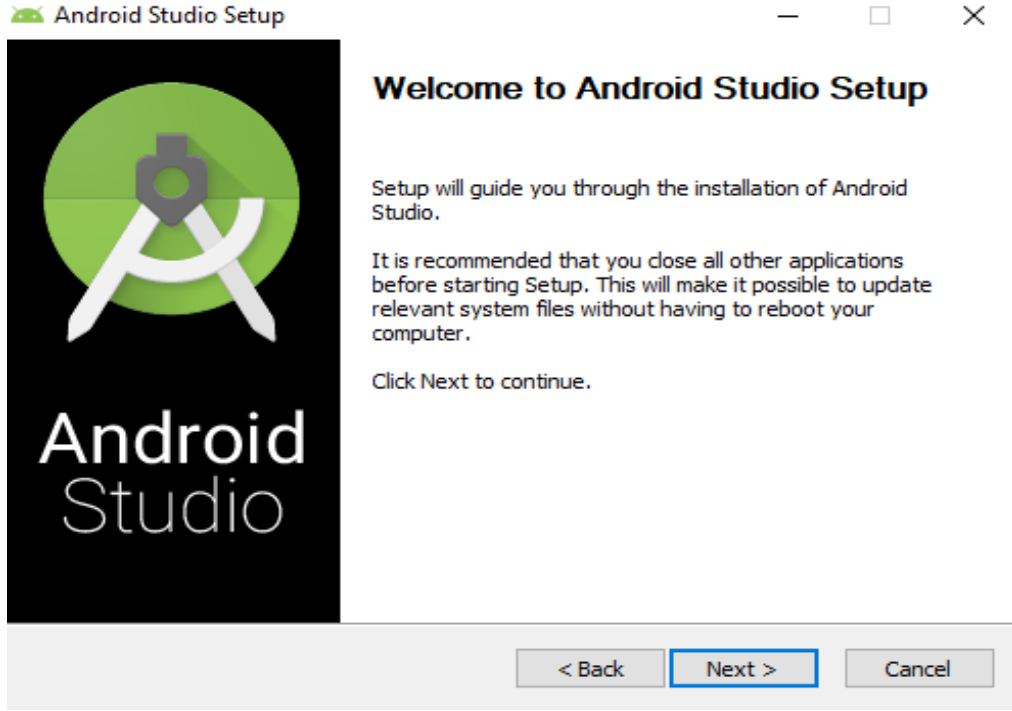
DOWNLOAD ANDROID STUDIO FOR WINDOWS

Şekil 28. Android Studio indirme

Sözleşmeyi kabul ederek indirme işlemini başlatıyoruz. İndirme işlemi tamamlandıktan sonra dosyayı açarak kurulumu başlıyoruz.

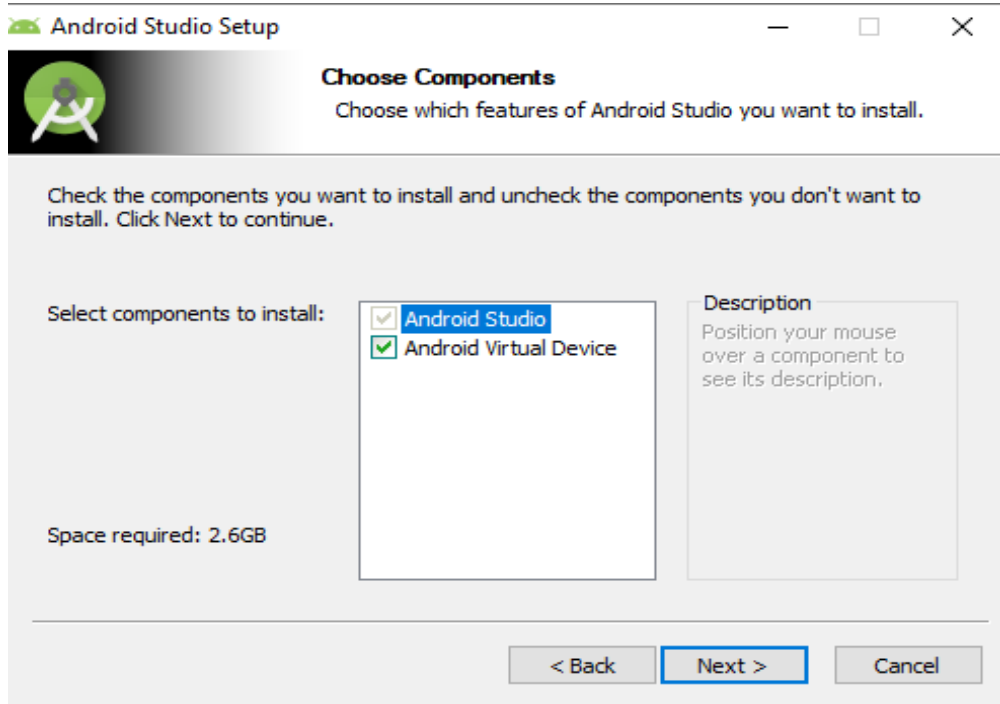
Kurulum dosyasının ilk sayfası Şekil 29’da gösterilmiştir.

Kurulum dosyasını açıp ‘Next’ tuşuna basarak bir sonraki adıma geçiyoruz.



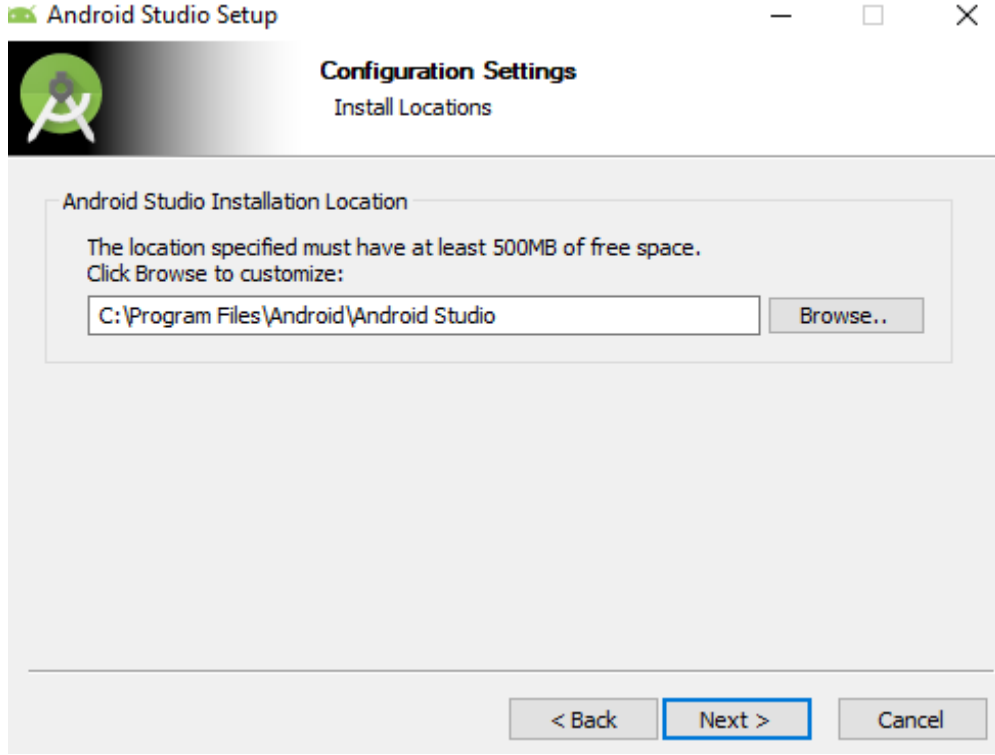
Şekil 29. Android Studio kurulum ilk adım

Şekil 30’da yüklenecek araçlar gösterilmiştir. Yükleme istediğimiz bileşenleri seçerek ‘Next’ tuşuna basıp bir sonraki adıma geçiyoruz.



Şekil 30. Android Studio kurulum seçenekleri

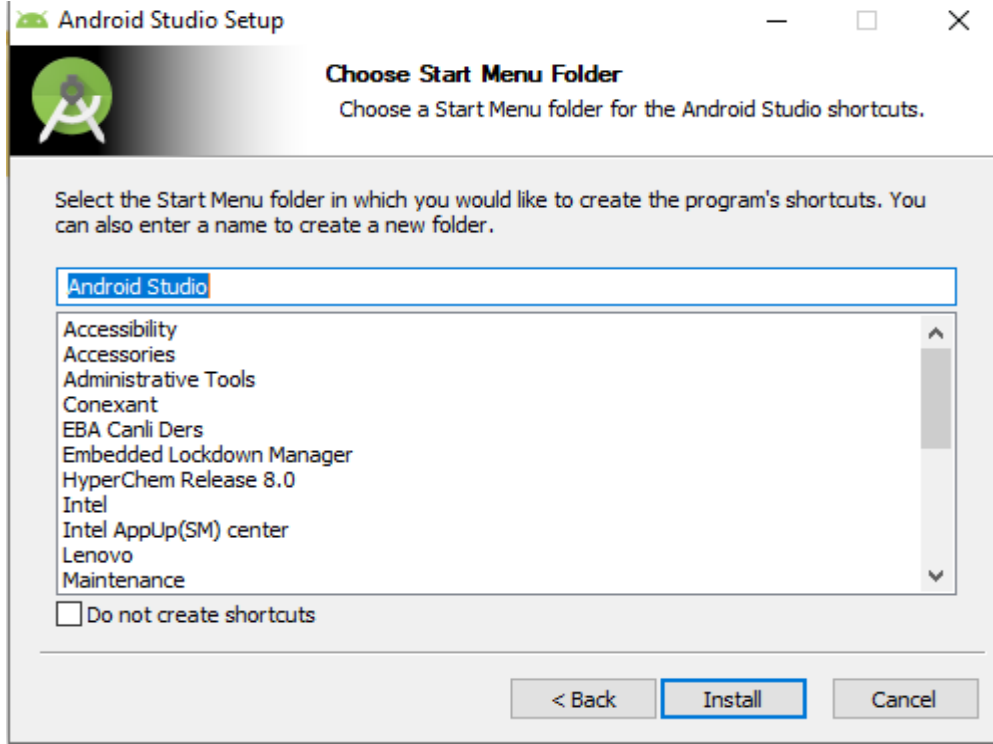
Uygulamanın nereye yükleneceği Şekil’31 de gösterilmiştir. Uygulamayı nereye yükleyeceğimizi seçip ve ‘Next’ diyerek bir sonraki adıma geçiyoruz.



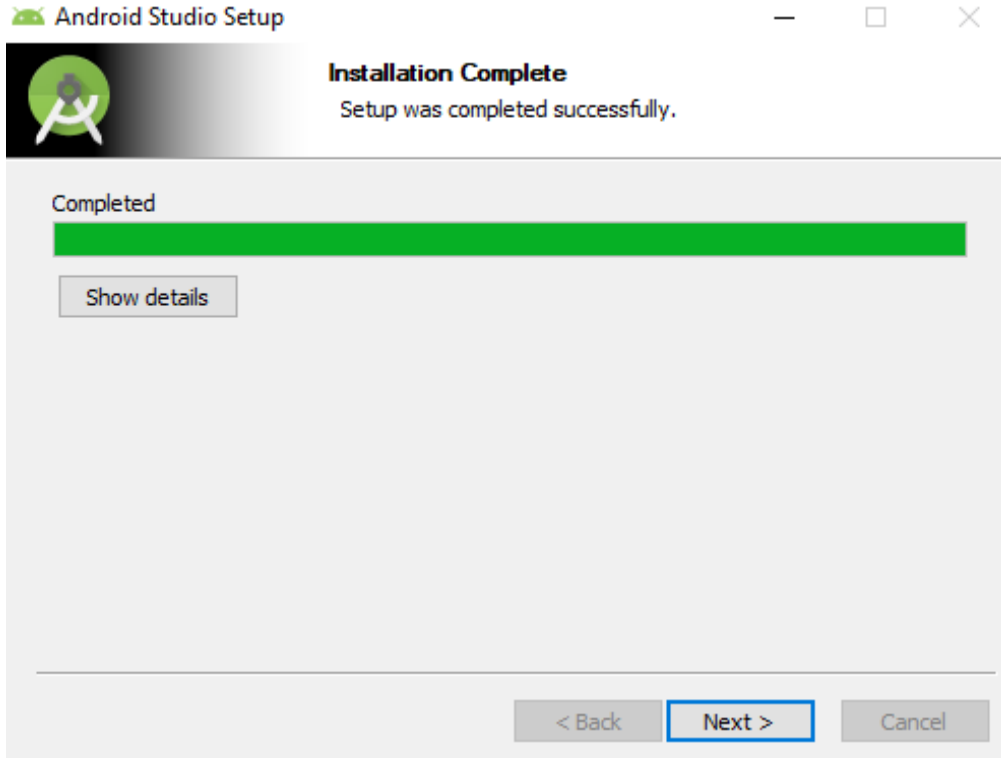
Şekil 31. Android Studio kurulum dizin seçimi

Android Studio'nun kurulumunun başlatıldığı adım Şekil 32'de gösterilmiştir.

Android Studio'nun "Başlat" çubuğunda kısayol olarak gözükmemesini istemiyorsak "Do not create shortcuts" seçeneğini işaretliyoruz ve "Install" butonuna basıyoruz. Ardından kurulum işlemi başlatılmış oluyor. Kurulum bittikten sonra "Next" butonuna tıklıyoruz. Çıkan pencereden "Finish" butonuna tıklıyoruz ve kurulumu sonlandırıyoruz.



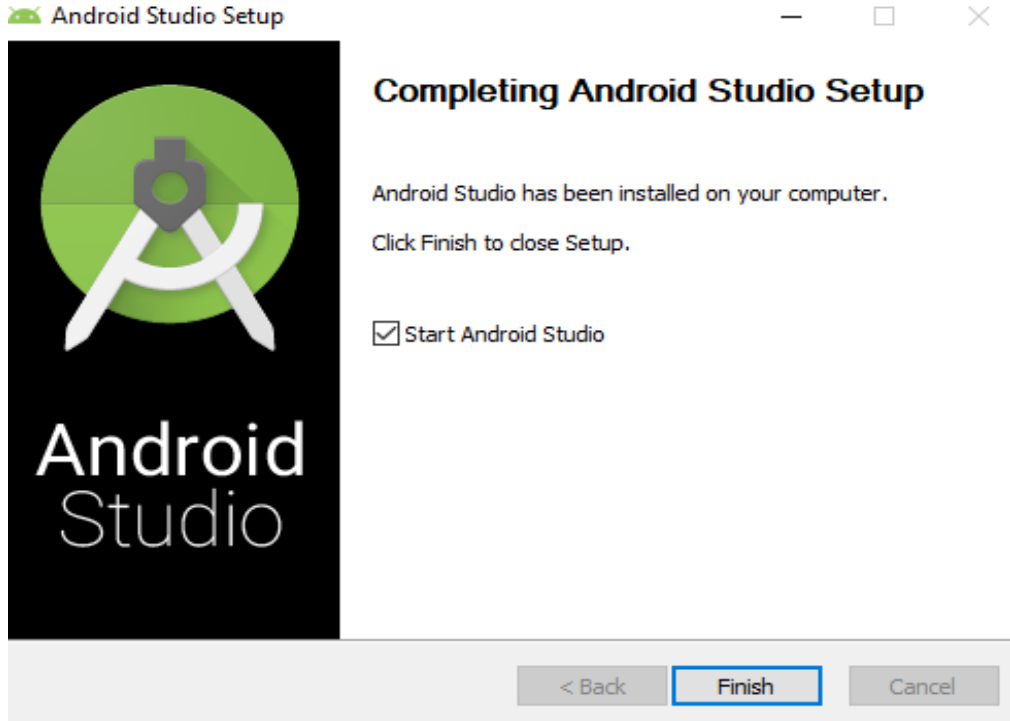
Şekil 32. Android Studio kurulumu yükleme



Şekil 33. Android Studio son kurulum adımı

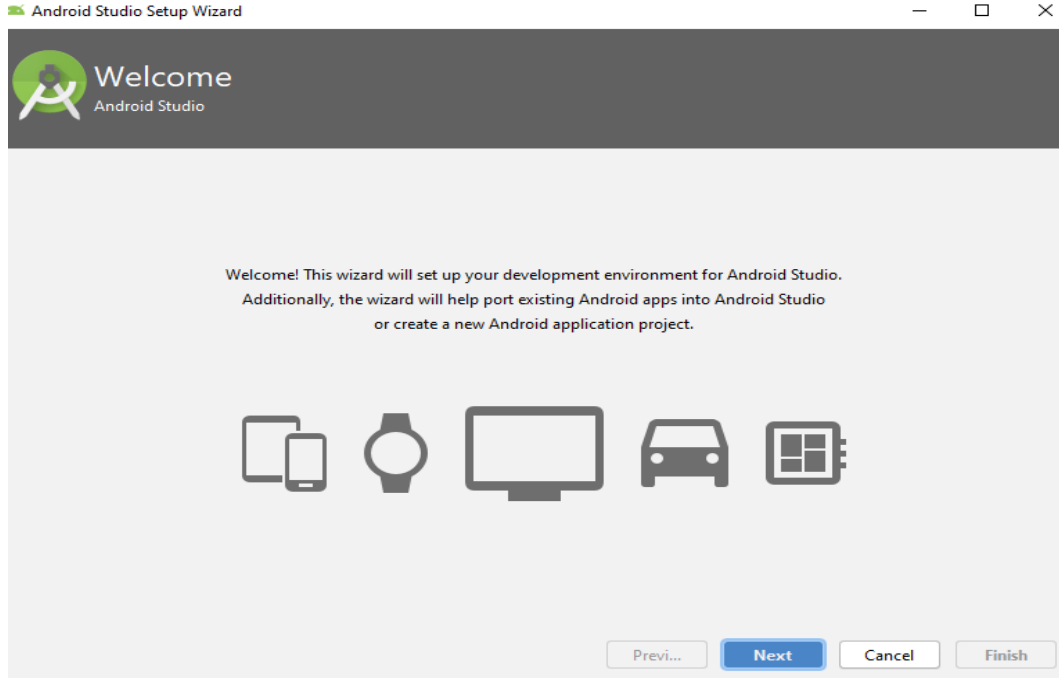
Şekil 33’de kurulumun son adımı verilmiştir.

Kurulum işlemi başarıyla gerçekleştirildi. Şimdi yeni bir proje oluşturalım.

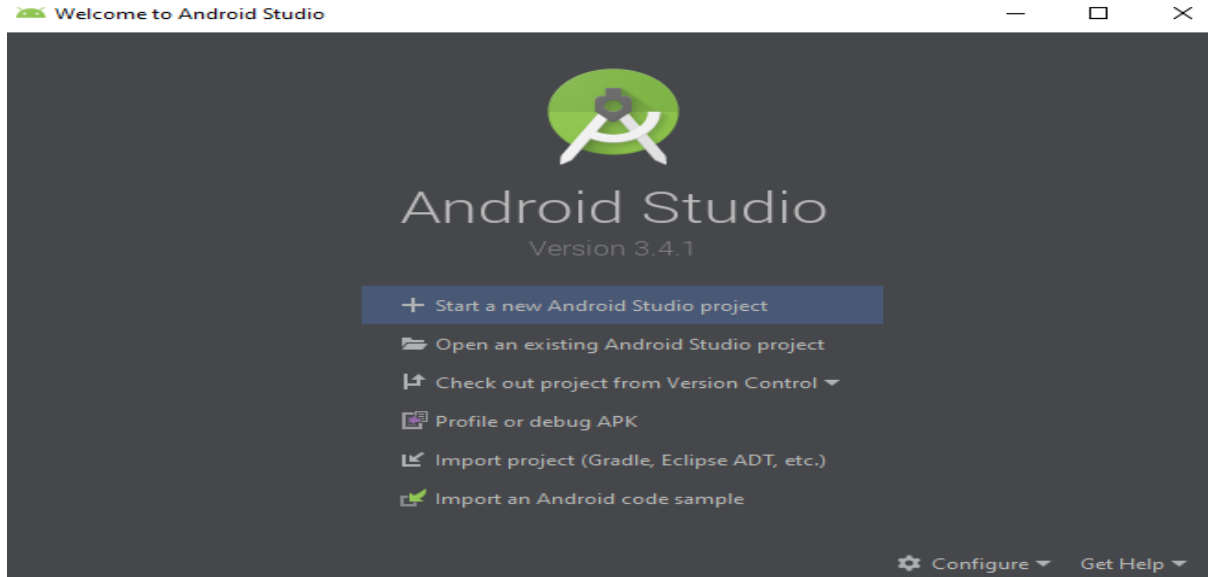


Şekil 34. Android Studio kurulum tamamlama adımı

Kurulum tamamlandıktan sonra bizi karşılayan sayfadan 'Next' diyerek giriş sayfasına geçiş yapıyoruz.

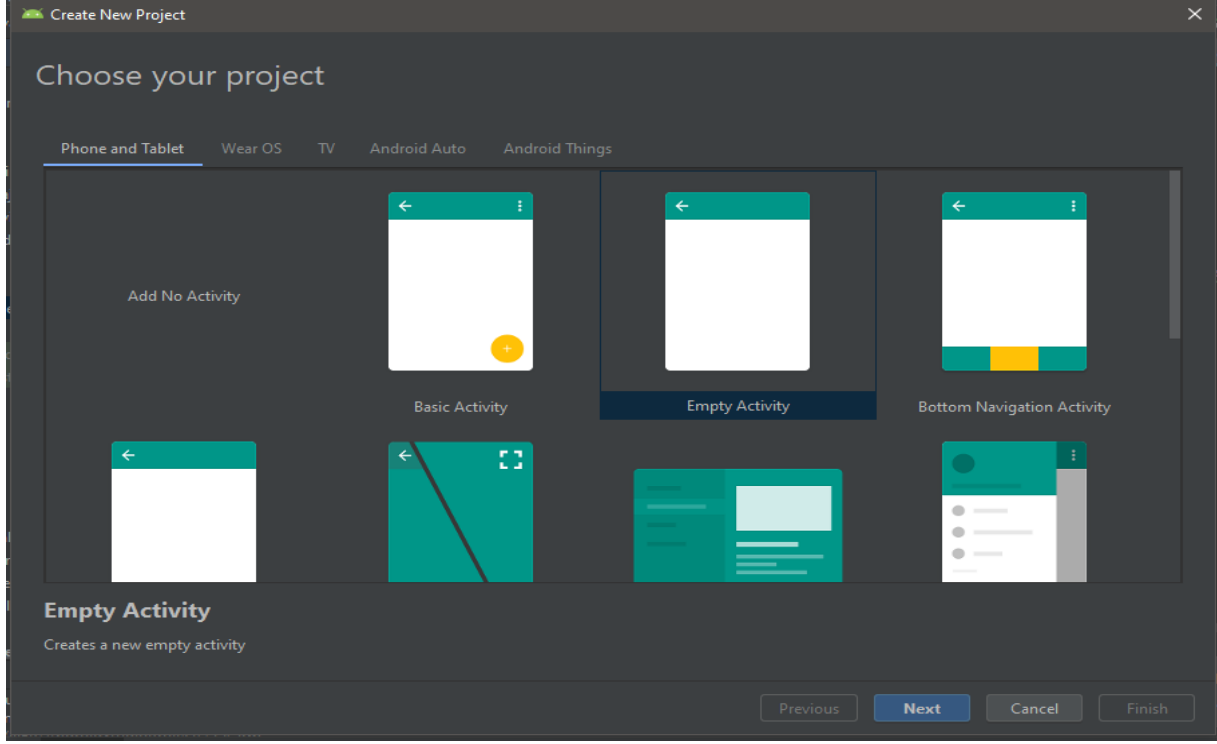


Şekil 35. Android hoş geldin sayfası



Şekil 36. Android Studio giriş sayfası

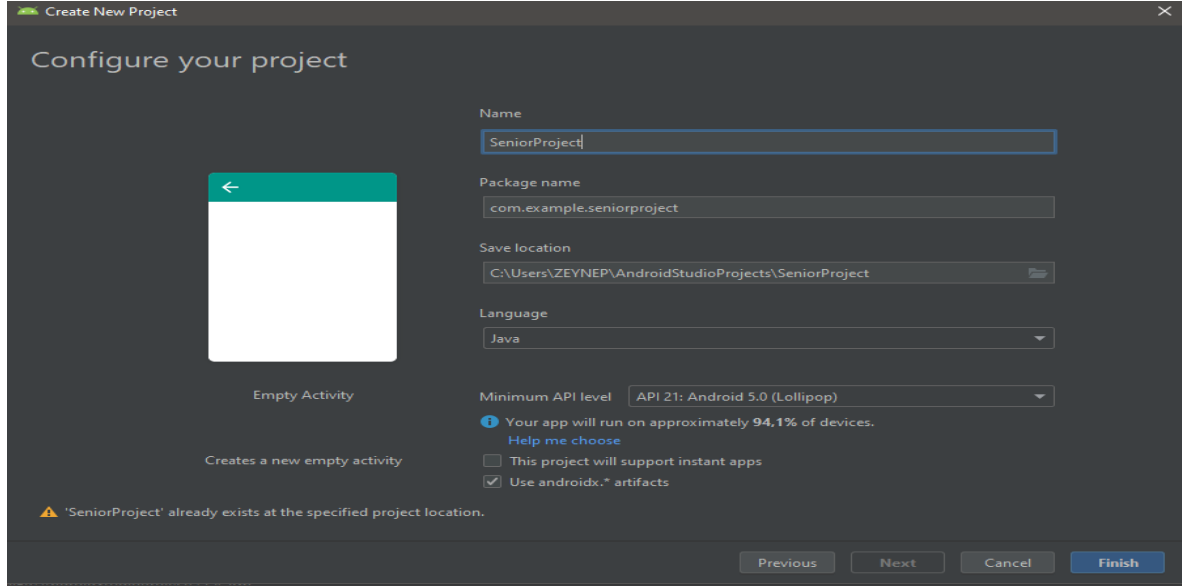
Android giriş sayfasından “Start a new Android Studio Project” seçeneğini seçerek yeni bir proje oluşturma adımlarına geçiyoruz.



Şekil 37. Android Studio template seçimi

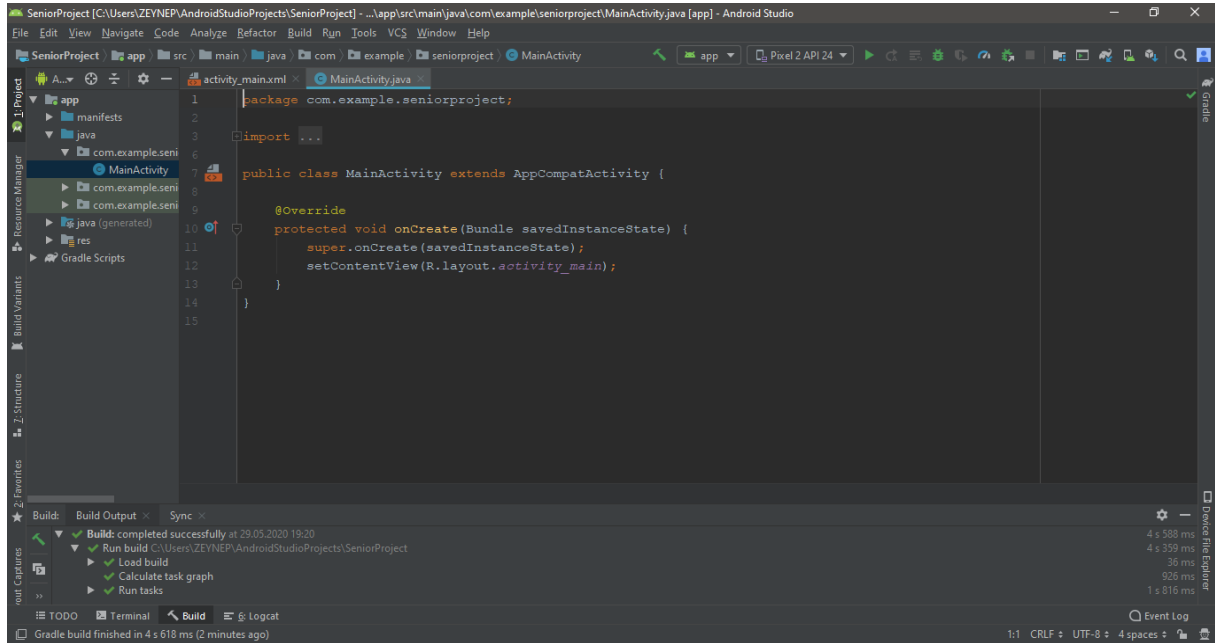
Şekil 37’de Android Studio’nun bize sunduğu templateler gösterilmiştir.

Bu ekranda Android bize template Activity’ler sunuyor. Empty Activity’i seçerek Next’e tıklıyoruz. İlerleyen zamanlar da projenin farklı ihtiyaçlarına göre templateler seçilebilir.



Şekil 38. Android Studio aktivite oluşturma adımı

Activity ve Layout'a isim verdiğimiz ekran Şekil 38'de gösterilmiştir. Bu adımda Activity ve Layout'lara isimlerini veriyoruz ve 'finish' butonuna tıklayarak projeyi oluşturuyoruz.



Şekil 39. Android Studio örnek bir proje

Kurulumu tamamlanmış örnek bir proje Şekil 39’de gösterilmiştir.

Projemiz oluşturuldu ve artık uygulamamızı geliştirmeye başlayabiliriz.

2.4.3. Android Studio’da Kullanılan Bileşenler

Mobil uygulamada ihtiyaçlar doğrultusunda Spinner, ListView, Volley, ve Gson teknolojileri kullanılmıştır. Gson kütüphanesi JSON verilerinin parse etme işlemleri için kullanılmıştır.

Volley

Volley, Android uygulamaları için ağı daha kolay ve en önemlisi daha hızlı hale getiren bir HTTP kütüphanesidir [6].

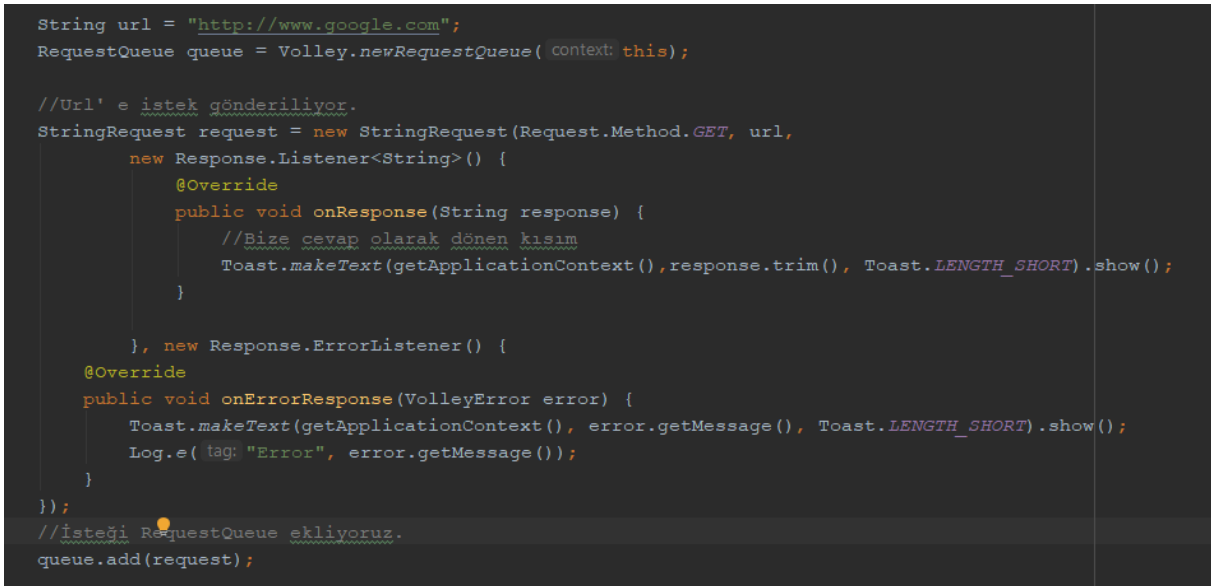
Volley aşağıdaki avantajları sunar:

- Ağ isteklerinin otomatik zamanlaması.
- Birden fazla eşzamanlı ağ bağlantısı.
- Standart HTTP önbellek tutarlılığı ile önbellekleme yapan şeffaf disk ve bellek yanıtı.
- İstek önceliği için destek.
- API iptal isteği mevcuttur. Tek bir isteği iptal edebilir veya iptal etmek için isteklerin kapsamalarını veya kapsamalarını belirleyebilirsiniz.
- Özelleştirme kolaylığı, örneğin, yeniden deneme ve geri çekilme için kullanılabilir.
- Kullanıcı arabiriminizi ağıdan eş zamansız olarak alınan verilerle doğru şekilde doldurmanızı kolaylaştırır.
- Hata ayıklama ve izleme araçları mevcuttur.
- Volley, bir arama sonuç sayfasını yapılandırılmış veri olarak almak gibi bir UI(User Interface)’i doldurmak için kullanılan RPC(Remote Procedure Call) tipi işlemlerde öne

çıkartır. Herhangi bir protokolle kolayca bütünleşir ve ham dizgiler, görüntüler ve JSON desteği ile kullanılabilir. İhtiyacınız olan özellikler için yerleşik destek sağlar. Şekil 29’de örnek bir GET isteği gösterilmiştir.

Volley kütüphanesini Android uygulamalarında kullanabilmek için Gradle dosyasına aşağıdaki kod satırının eklenmesi gerekmektedir. build.gradle(Module: app) yazan yere tıklayıp kod satırını ekliyoruz.

```
dependencies {  
    ...  
    implementation 'com.android.volley:volley:1.1.1'  
}
```

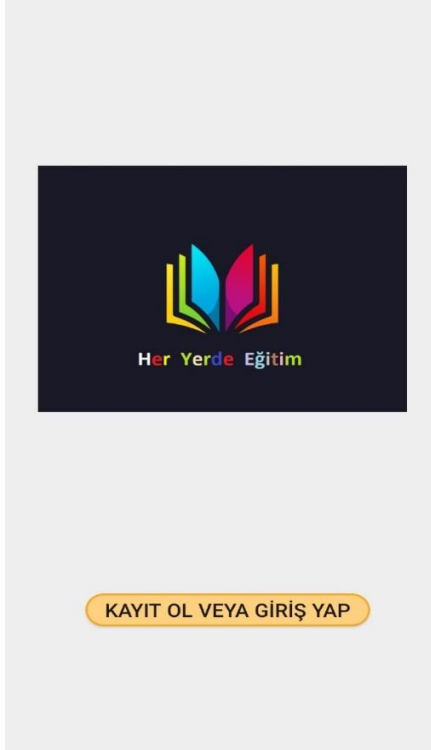
A screenshot of a code editor showing Java code that uses the Volley library to make a GET request to 'http://www.google.com'. The code includes a RequestQueue, a StringRequest, and two listeners: one for successful responses (onResponse) and one for errors (onErrorResponse). The onResponse method shows a toast message with the response, and the onErrorResponse method shows a toast message with the error message. The code is written in a dark-themed editor with syntax highlighting.

```
String url = "http://www.google.com";  
RequestQueue queue = Volley.newRequestQueue( context: this);  
  
//Url' e istek gönderiliyor.  
StringRequest request = new StringRequest(Request.Method.GET, url,  
    new Response.Listener<String>() {  
        @Override  
        public void onResponse(String response) {  
            //Bize cevap olarak dönen kısım  
            Toast.makeText(getApplicationContext(), response.trim(), Toast.LENGTH_SHORT).show();  
        }  
    }, new Response.ErrorListener() {  
        @Override  
        public void onErrorResponse(VolleyError error) {  
            Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_SHORT).show();  
            Log.e( tag: "Error", error.getMessage());  
        }  
    }  
));  
//İsteği RequestQueue ekliyoruz.  
queue.add(request);
```

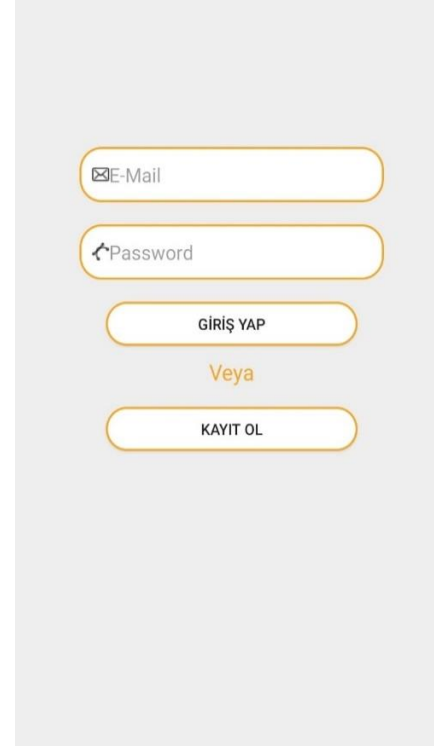
Şekil 40. Volley kullanım örneği

2.4.4. Mobil Uygulama Geliştirme Süreci

Uygulama açıldığında kullanıcılar uygulama ana ekranına yönlendirilmektedir. Buradan kayıtlı kullanıcıların giriş yapabilmesi için veya kayıtlı olmayan kullanıcıların sisteme kayıt olabilmesi için uygulamanın giriş ekranına yönlendirilmektedir. Uygulamanın ana ekranı Şekil 41’de, giriş ekranı ise Şekil 42’de gösterilmiştir.

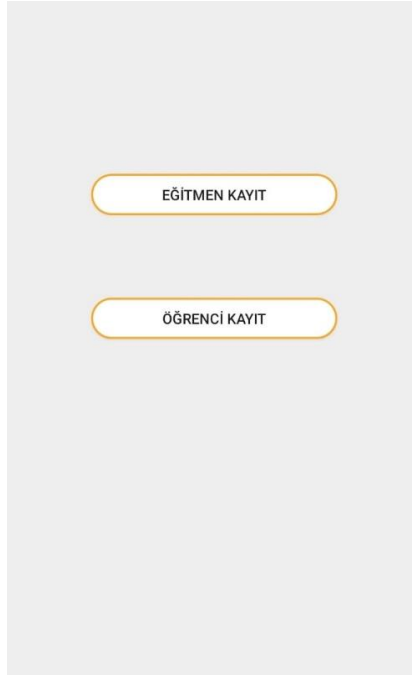


Şekil 41. Mobil Uygulama ana ekranı



Şekil 42. Mobil Uygulama giriş ekranı

Uygulamamızda Eđitmen ve Öğrenci olmak üzere iki farklı kullanıcı tipi bulunmaktadır. Uygulamaya kayıtlı olmayan kullanıcılar Kayıt Ol Butonuna tıkladıktan sonra Kayıt Seçme ekranından Eđitmen veya Öğrenci olarak kayıt tipini seçtikten sonra ilgili kullanıcının kayıt sayfasına yönlendirilmektedir. Şekil 43'te kayıt seçme ekranı gösterilmiştir.



Şekil 43. Mobil uygulama kayıt seçme ekranı

ÖĞRENCİ KAYDI

Adınız

Soyadınız

Doğum Tarihiniz

Okul Bilgisi

☐ Kadın ☐ Erkek

Adresiniz



FOTOĞRAF SEÇ

Mail Adresiniz

Şifreniz

KAYIT OL

Şekil 44. Mobil uygulama öğrenci kaydı

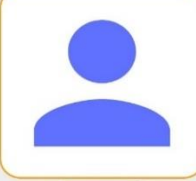
EĞİTMEN KAYDI

Adınız

Soyadınız

Doğum Tarihiniz

☐ Kadın ☐ Erkek



FOTOĞRAF SEÇ

Branşınızı Seçiniz

Matematik Matematik

Ücret Talebiniz

Adresiniz

Telefon Numaranız

Mail Adresiniz

Şifreniz

KAYIT OL

Şekil 45. Mobil uygulama öğretmen kaydı

Şekil 44 ve Şekil 45'te farklı kullanıcı tiplerinin uygulamaya kayıt sayfaları gösterilmiştir. Öğrenci ve öğretmen kayıt olurken Post metodu kullanılarak gerekli bilgiler RESTful API aracılığı ile veri tabanına kaydedilmektedir..

Mobil Uygulamanın Öğrenci Tarafı



Şekil 46. Mobil uygulama öğrenci ana sayfası

Öğrenci uygulamaya başarılı bir şekilde giriş yaptıktan sonra Şekil 47’de ki öğrenci ana sayfasına yönlendirilmektedir.



Şekil 47. Mobil uygulama öğrenci profili

Öğrenci bu ana sayfadaki menünün profil kısmından profil bilgilerini güncelleyebilir, aldığı dersleri görüntüleyebilir ve aldığı ders, eğitmen tarafından onaylandıktan sonra ilgili eğitmenle mesajlaşabilir.

ÖĞRENCİ DERS ALMA

Almak İstediğin Ders

Matematik

Dersin Yapılacağı Şehir

Şehir

EĞİTMENLERİ GÖRÜNTÜLE

Şekil 48. Mobil uygulama öğrenci ders talebi

Kayıt olan öğrenci, menüdeki ders talebi seçeneğinden Şekil 48 ‘de ki Ders talep etme sayfasına yönlendirilmektedir. Bu sayfada almak istediği dersi ve dersin yapılacağı adres bilgisini girdikten sonra, uygulamada bu kriterlerle kayıt olan öğretmenler boş gün bilgisine göre öğrenci karşısında listelenmektedir. Ekranda öğrencinin alabileceği dersler veri tabanından çekilerek Spinner ile listelenmiştir.

UYGUN EĞİTMENLER
Teacher : Ahmet Aras Adres : Istanbul Ücret : 150 Tarih :2020-05-31 Saat : 15:00:00 Süre : 6
Teacher : Ali Kara Adres : Istanbul Ücret : 150 Tarih :2020-05-29 Saat : 15:00:00 Süre : 2
Teacher : Serhat Hakan Adres : Istanbul Ücret : 180 Tarih :2020-06-07 Saat : 18:00:00 Süre : 1

Şekil 49. Mobil uygulama öğrenci kriterlerine uygun eğitmenler

Buradaki eğitmen bilgileri RESTful API ile veri tabanından çekilerek listelenmiştir. Eğitmenlerin listelenmesinde listView kullanılmıştır. Eğitmenler Şekil 50’de ki gibi listView içerisinde listelenmiştir. Öğrenci listelenen eğitmenlerin herhangi birinin üzerine tıkladığında ilgili eğitmen ile ilgili daha detaylı bilgilerin bulunduğu sayfaya yönlendirilmektedir.

EGİTMEN BİLGİLERİ

Adı : Ahmet Aras
Okulu : YTÜ
Doğum Tarihi : 1986-05-04
Adres : İstanbul
Ücret : 150
Email : ahmet@gmail.com
Puan Ortalaması : 8

EGİTMEN YORUMLARI

Student : Azra Aydemir
Yorum : Verimli geçti
Puan : 8
Tarih :2020-05-26

Student : Irem naz Gunhan
Yorum : Anlatımı güzeldi.
Puan : 9
Tarih :2020-06-16

Student : Hira Ilbasiniz
Yorum : Sadece biraz hızlı konuşması var ama gayet iyiydi.
Puan : 8
Tarih :2020-05-20

Student : Aslı Aslan
Yorum : Konuya hakim..akıcı bir anlatımı var.
Puan : 9
Tarih :2020-05-31

Student : Ali Aras
Yorum : Gayet başarılı bir öğretmen.

Ders Ortamını Seçin

Eğitmenin Evinde

Eğitmenin Evinde

DERS TALEBİ OLUŞTUR

Şekil 50. Mobil uygulama eğitmen detay

Şekil 50’ de görüldüğü gibi bu sayfada, öğrencinin seçtiği eğitmenin mezun olduğu okul, doğum tarihi, adresi, ders için talep ettiği ücret, e-mail adresi eğitmen bilgileri başlığı altında yer almaktadır. Ayrıca ilgili eğitmenden daha önce ders alan öğrencilerin eğitmene verdiği puanların ortalaması alınarak eğitmen bilgilerine eklenmiştir. Eğitmen yorumları başlığı altında ise, daha önce eğitmenden ders almış olan öğrencilerin eğitmene yaptıkları yorumlar ve verdikleri puanlar listelenmiştir. Buradaki eğitmen bilgileri ve yorumlar RESTful API ile veri tabanından çekilerek listelenmiştir. Eğitmen bilgilerinin ve yorumların listelenmesinde listView kullanılmıştır. Öğrenci eğitmen seçimini tamamladıktan sonra, dersin yapılacağı mekan bilgisini seçmelidir. Mekan

bilgisi seçimini tamamlayan öğrenci Ders Talebi Oluştur butonuna tıklayarak ders talebi oluşturulabilmektedir.

Aldığım Dersler
Eğitmen Adı : Mustafa Yılmaz Ders Adı : Matematik Mekan : Kafe Tarih :2020-05-29 Saat : 16:30:00 Süre : 1 Ücret : 80 Durum : Onaylandı
Eğitmen Adı : Ahmet Aras Ders Adı : Fen Bilgisi Mekan : Eğitmenin Evinde Tarih :2020-06-30 Saat : 13:00:00 Süre : 2 Ücret : 150 Durum : Beklemede

Şekil 51. Mobil uygulama alınan dersler

Öğrencinin oluşturduğu ders talepleri Şekil 51’de ki gibi profilinde aldığım dersler kısmında listelenmektedir.

Oluşturulan yeni ders talebinin durumu ilk olarak Beklemede olacak şekilde Api tarafından otomatik olarak atanmaktadır . Eğitimci oluşturulan ders talebini onaylayabilir veya reddedebilir. Eğitimci ders talebini onaylarsa dersin durumu Onaylandı durumuna, reddederse Reddedildi durumuna düşmektedir



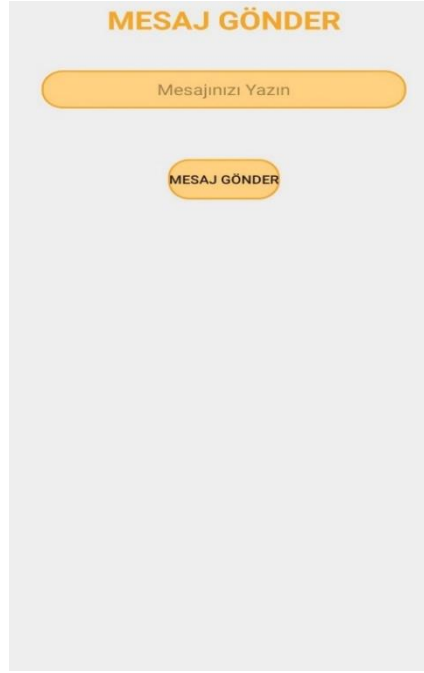
Şekil 52.Mobil uygulama eğitimci mesajlaşma



Şekil 53.Mobil uygulama eğitimci mesaj listesi

Ders ile alakalı son anda gelişen olumsuzluklar veya herhangi bir sebeple haberleşme gereği duyan eğitimci ve öğrenci için uygulamaya mesajlaşma özelliği eklenmiştir. Öğrencinin oluşturmuş olduğu ders talebi eğitimci tarafından onaylanırsa eğitimci ve öğrenci arasında

mesajlaşma özelliği aktif hale gelmektedir. Şekil 52’de görüldüğü gibi, öğrenci profilindeki Mesajlar kısmına tıkladığında kendisine mesaj gönderen eğitimler listelenmektedir. Öğrenci listelenen eğitimlerin üzerine tıkladığında ilgili eğitim ile yaptığı mevcut konuşma Şekil 53’te gibi görüntülenmektedir.



Şekil 54. Mobil uygulama öğrenci mesaj gönderme

Öğrenci Şekil 54’e ki gibi, mesajın üzerine tıklayarak mesaj oluşturma sayfasına yönlendirilmekte ve kendisine gelen mesajlara yanıt oluşturabilmektedir.

Şekil 55. Mobil uygulama eğitimci değerlendirme

Uygulamada, ders eğitimci tarafından onaylanıp ders alma işlemi gerçekleştirildikten sonra öğrencinin aldığı ders ile ilgili eğitimci değerlendirmesi zorunlu tutulmuştur. Bu sayede öğrenciler eğitimci seçiminde sistemde ders almış diğer öğrencilerin yaptığı değerlendirmeler sonucu eğitimcilerle ilgili fikir sahibi olabileceklerdir.

Öğrenciler profilindeki Aldığım Dersler Kısmından ilgili derse tıklayarak, Şekil 55'te bulunan sayfadan aldığı dersle ilgili eğitimci değerlendirebilmektedir.

Mobil Uygulamanın Eđitmen Tarafı

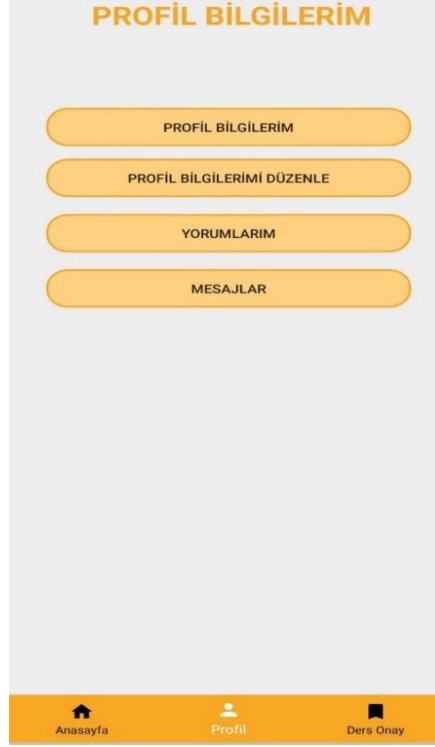
Eđitmen uygulamaya başarılı bir şekilde giriş yaptıktan sonra Şekil 56'da ki eđitmen ana sayfasına yönlendirilir.

The screenshot shows the 'Hoşgeldiniz !' (Welcome!) screen for a teacher. It features a 'Name' label at the top. Below it is a section titled 'BOŞ GÜN BİLGİMİ GÜNCELLE' (Update my empty day information). This section contains a 'Ders Süresi' (Lesson Duration) label and a text input field for 'Verebileceğiniz Ders Saat Sayısı' (Number of lesson hours you can give). Below this is a 'TAKVİMİ AÇ' (Open Calendar) button, followed by a 'Tarih' (Date) input field. Then there is a 'SAATİ AÇ' (Open Hour) button, followed by a 'Saat' (Hour) input field. Below these are two buttons: 'GÜNCELLE' (Update) and 'ÇIKIŞ' (Exit). At the bottom, there is an orange navigation bar with three icons and labels: 'Anasayfa' (Home), 'Profil' (Profile), and 'Ders Onay' (Lesson Approval).

Şekil 56. Mobil uygulama eđitmen ana sayfa

Eđitmen Şekil 56'da görüldüğü gibi ana sayfasında boş gün bilgisini güncelleyebilmektedir. Eđitmenler güncel olan boş gün bilgileriyle, ders talep etmek isteyen öğrencilerin filtreleri sonucu listelenebilmektedir. Bu yüzden uygulamada eđitmenin boş gün bilgisi güncel olmalıdır.

Bu ekranda eğitimci, kendisine sunulan takvimden boş gününü, saatten ise boş olduğu saatleri sisteme kaydedebilmektedir. Ders süresi kısmına ise kaç saatinin boş olduğu bilgisini girmektedir.



Şekil 57. Mobil uygulama eğitimci profil

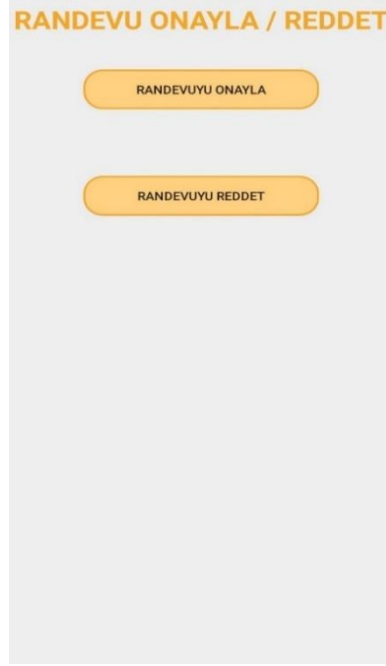
Eğitimci Şekil 57’de ana sayfasındaki menünün profil kısmından profil bilgilerini güncelleyebilir, kendisine yapılan yorumları listeleyebilir ve kendisine gelen ders taleplerini onayladıktan sonra ilgili öğrenciyle mesajlaşabilmektedir.

RANDEVU LİSTEM	
Öğrenci : Hira Ilbasiniz	
Tarih :2020-05-31	
Saat : 15:00:00	
Mekan :Oğrencinin Evi	
Durum : Beklemede	
Öğrenci : Ali Aras	
Tarih :2020-05-31	
Saat : 15:00:00	
Mekan :Kafe	
Durum : Beklemede	
Öğrenci : Aslı Aslan	
Tarih :2020-05-31	
Saat : 15:00:00	
Mekan :Oğrencinin Evi	
Durum : Beklemede	
Öğrenci : Azra Aydemir	
Tarih :2020-05-31	
Saat : 15:00:00	
Mekan :Dershane/Etüt Merkez	
Durum : Beklemede	

Şekil 58. Mobil uygulama eğitimci mesaj listesi

Öğrenciler boş gün bilgisine göre eğitime ders talebi gönderebilmektedir.

Eğitmen Şekil 58’ de ki gibi menüdeki Ders Onay kısmından ise kendisine gelen ders taleplerini listeleyebilmektedir. Öğrenci kendisinden ders talep ettiğinde dersin durumu otomatik olarak Beklemede olmaktadır.



Şekil 59. Mobil uygulama eğitimci randevu onay/red

Şekil 59’da ki ekrandan eğitimci kendisine gelen ders talepleri listesinde istediği dersi Onaylama veya Reddetme imkanına sahiptir. Eğitimci kendisine gelen taleplerini onayladığı durumda ilgili öğrenci ile mesajlaşma özelliği aktif olmaktadır. Eğitimci kendi profilindeki mesajlar kısmında mesajlarını listeleyebilmekte ve ilgili öğrencilere mesaj gönderebilmektedir.



Şekil 60. Mobil uygulama eğitimci mesajlaşma Şekil 61. Mobil uygulama eğitimci mesaj listesi

Eğitimci profilindeki Mesajlar kısmına tıkladığında kendisine mesaj gönderen öğrenciler listelenmektedir (Şekil 60). Eğitimci listelenen öğrenci isimlerinin üzerine tıkladığında ilgili öğrenci ile yaptığı mevcut konuşma görüntülenmektedir. Eğitimci mesajın üzerine tıklayarak mesaj oluşturma sayfasına yönlendirilmekte ve kendisine gelen mesaja yanıt oluşturabilmektedir (Şekil 61).

MESAJ GÖNDER

Mesajınızı Yazın

MESAJ GÖNDER

Şekil 62. Mobil uygulama eğitimci mesaj gönderme

Öğrenci mesajın üzerine tıkladığında Şekil 62’de ki mesaj oluşturma sayfasına yönlendirilmekte ve kendisine gelen mesajlara yanıt oluşturabilmektedir.

BÖLÜM 3

KAYNAKLAR

1. İnternet: “REST API Nedir?”,
<https://burcualtinok.com.tr/blog/rest-api-nedir/>
2. İnternet: “Http İstek Metodları”,
<https://medium.com/@gokhan.746/http-i%C4%B1-http-request-methods-8f43d4d9c4c6>
3. İnternet: “REST Mimari Özellikleri”,
<http://cagataykiziltan.net/rest-restful-soap/>
4. İnternet: “PhpMyAdmin Nedir?”,
<https://wmaraci.com/nedir/phpmyadmin>
5. İnternet: “Postman Nedir?”,
<https://medium.com/postman-t%C3%BCrkiye/postman-nedir-622be8afef2e>
6. İnternet: Android Studio Volley Kütüphanesi
<https://medium.com/@halilozel1903/android-volley-kullan%C4%B1m%C4%B1-8ee56b6762f>