

**KARADENİZ TEKNİK ÜNİVERSİTESİ
OF TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**



CRYPTOLIVE WEB/MOBİL PROJESİ

TASARIM PROJESİ

**Yusuf Çolak
Fatih Demirtaş**

2022-2023 GÜZ DÖNEMİ

**KARADENİZ TEKNİK ÜNİVERSİTESİ
OF TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**CRYPTOLIVE WEB/MOBİLE
PROJESİ**

TASARIM PROJESİ

**Yusuf Çolak
Fatih Demirtaş**

Bu projenin teslim edilmesi ve sunulması tarafımdan uygundur.

Danışman: Öğr. Gör Selda ATALAR

.....

2022-2023 GÜZ DÖNEMİ



IEEE Etik Kuralları IEEE Code of Ethics



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

Bu çalışma, pandemi döneminde müzisyenlerin, sanatçıların ve yeni müzik yeteneklerinin, müziği sevdalılarıyla buluşturabilmesi adına geliştirilmiştir. Küresel salgın, geleneksel canlı performanslara olan erişimi kısıtladı, ancak bu platform, sanatçıların ve müzik tutkunlarının bir araya gelmesine, etkileşimde bulunmasına ve paylaşmasına olanak tanıyan yeni bir dijital deneyim sunuyor.

Pandemi sona erse bile, bu platform, müzik dünyasında sürdürülebilir bir dijital etkileşim modelinin ne kadar önemli olduğunu vurgulamaktadır. Yenilikçi teknolojiler, müzisyenlere eserlerini paylaşma ve dinleyicilere daha geniş bir kitleye erişme fırsatı sunarak, müziğin evrenselliğini daha da pekiştirmektedir. Bu sayede, gelecekteki müzik sahneleri, dijitalleşme ve etkileşim konularında öncü olacak, sanatçılar da yaratıcılıklarını daha geniş bir izleyici kitlesiyle paylaşma imkanı bulacaklardır.

Çalışma konusunun belirlenmesinde ve çalışmanın hazırlanma sürecinin her aşamasında bilgilerini, tecrübelerini ve değerli zamanlarını esirgemeyerek bize her fırsatta yardımcı olan değerli hocamız Sayın Öğr. Gör Selda ATALAR'a teşekkür ve minnetimizi sunarız.

Yusuf ÇOLAK
Fatih Demirtaş
Trabzon, 2022

İÇİNDEKİLER

Sayfa No

| | |
|--|------|
| IEEE ETİK KURALLARI | II |
| ÖNSÖZ | III |
| İÇİNDEKİLER | IV |
| ÖZET | VI |
| ŞEKİLLER DİZİNİ | VIII |
| 1. GENEL BİLGİLER | 1 |
| 1.1. Giriş | 1 |
| 1.2. Amaçlar | 2 |
| 1.2.1. Sanatçıların İlerlemesini Hızlandırmak | 2 |
| 1.2.2. Topluluğa Hızlı Erişim | 2 |
| 1.2.3. Destek ve Bağış İmkanları: | 2 |
| 1.2.4. Kripto Para ile Hızlı Ödemeler | 2 |
| 1.2.5. Takip Edilebilir İşlemler | 2 |
| 12.6. Küresel Duyuru ve Tanıtım | 2 |
| 1.2.7. Ödeme Yöntemi ve Güvenlik | 3 |
| 2. Kullanılacak Teknolojiler | 3 |
| 2.1. Web Geliştirme | 3 |
| 2.2. Canlı Yayın Protokolleri | 3 |
| 2.3. Veri tabanı | 3 |
| 2.4. Sunucu ve İstemci İletişimi | 3 |
| 2.5 Güvenlik ve Kimlik Doğrulama | 3 |
| 2.6. Ödeme Yöntemi | 4 |
| 2.7. Mobil Uygulama | 4 |
| 3. Proje Oluşturma | 5 |
| 3.1. Veri tabanı ve JPA Entegrasyonu | 5 |
| 3.2. Controlller ve Servis Katmanlarının Oluşturulması | 8 |
| 3.3. React.JavaScript Front-end Kütüphanesi Kullanımı | 8 |
| 3.4. Güvenlik ve Kimlik Doğrulama | 9 |
| 3.5. Uygulama Testleri ve Hata Ayıklama | 9 |

| | |
|--|----|
| 3.6. Kripto Para ile Bağış Sistemi | 9 |
| 3.7. React Native | 10 |
| 3.8. Solidity Nedir?..... | 12 |
| 4. YAPILAN ÇALIŞMALAR..... | 13 |
| 4.3 İhtiyaç Analizinin Yapılması..... | 13 |
| 5. Proje Tasarımı | 14 |
| 6. KAYNAKLAR..... | 18 |
| STANDARTLAR ve KISITLAR FORMU..... | 19 |

ÖZET

Bu çalışma, 2019'da başlayan ve küresel çapta etkiler yaratan COVID-19 pandemisinin müzik endüstrisi üzerindeki etkilerini incelemektedir. Dudu, Öğüt ve Denizci'nin 2022 tarihli "Türkiye'de Müzik Emeginin Durumu" başlıklı araştırması, pandeminin Türkiye'deki müzik emeği üzerinde olumsuz etkilerini ortaya koymaktadır. Pandemi, müzisyenlerin güvencesiz çalışma koşulları ve gelir kayıplarıyla mücadele etmelerine neden olmuştur. Canlı performansların ve konser turnelerinin iptali, müzik endüstrisindeki yaratıcı emek gücünü olumsuz etkilemiştir.

Araştırma, Türkiye'deki müzik endüstrisinin pandemi öncesinde başlayan platformlaşma olgusundan etkilendiğini vurgulamakta ve pandemi süreciyle birlikte platformlaşmanın endüstri içindeki rolünün daha da önem kazandığını belirtmektedir. Canlı müzik performanslarının ve konser turnelerinin kapatılması, müzisyenlerin gelir kaynaklarını önemli ölçüde azaltmıştır. Kapanmanın en çok etkilediği alanlar arasında canlı müzik performanslarının yapıldığı sahneler, müzisyenler, tiyatro sanatçıları, tiyatro salonları, sinema salonları ve film festivalleri bulunmaktadır.

Araştırma, müzik endüstrisindeki yaratıcı emek gücünün, sahnelerin ve konser turnelerinin kapatılmasıyla önemli ölçüde olumsuz etkilendiğini ortaya koymaktadır. Türkiye'de, dünya genelinde olduğu gibi, müzisyenlerin gelirlerinin büyük bir kısmını canlı performanslardan elde ettiği bilinmektedir. Araştırma ayrıca, müzisyenlerin doğrudan dinleyicilere ulaşabilecekleri, canlı müzik yapabilecekleri bir platform eksikliğine dikkat çekmektedir.

Araştırmanın amacı, pandeminin müzik endüstrisindeki yaratıcı emek gücü üzerindeki etkilerini anlamak ve benzer durumlar için daha etkili destek mekanizmaları oluşturmak olarak belirlenmiştir. Sanatçıların ilerlemesini hızlandırmak, topluluğa hızlı erişim sağlamak, destek ve bağış imkanları sunmak, kripto para ile hızlı ödemeleri mümkün kılmak, takip edilebilir işlemler sağlamak, küresel duyuru ve tanıtım yapmak gibi amaçlar doğrultusunda bir proje önerisi sunulmuştur.

Projenin teknik altyapısı, web geliştirme için HTML, CSS, JavaScript (React), Node.js, Python, Java (Spring Boot); canlı yayın protokolleri için WebRTC, RTMP, HLS; veri tabanı için PostgreSQL; sunucu ve istemci iletişimi için WebSocket teknolojisi; güvenlik ve kimlik doğrulama için OAuth 2.0 protokolü ve JWT; ödeme yöntemi için Solidity ve Metamask; mobil uygulama için React Native kullanılması öngörülmektedir.

Çalışmanın sonuçları, müzik dünyasında yer almak isteyen sanatçılara daha geniş bir görünürlük ve destek sağlamayı amaçlayan bir projenin potansiyel başarılarına odaklanmaktadır. Ayrıca, kripto paraların kullanımının pandemi sonrası çevrimiçi bağışlar ve ödeme yöntemleri için gelecekte daha da yaygınlaşabileceğine dair önemli tespitler içermektedir.

ŞEKİLLER DİZİNİ

Sayfa No

| | |
|---|----|
| Şekil 1. RTMP sunucusunun çalışma mantığı | 4 |
| Şekil 2. RTMP sunucusunun CDN ile yayınlanma mantığı..... | 5 |
| Şekil 3. Ana Sayfa | 14 |
| Şekil 4. Takip Edilenler | 15 |
| Şekil 5. Side Bar | 16 |
| Şekil 6. Kullanıcı Profili | 17 |

1. GENEL BİLGİLER

1.1. Giriş

2019 yılında başlayan ve küresel bir çapta etkiler yaratan COVID-19 pandemisi, birçok sektöründen etkilemiştir. Literatür taramamızda, pandemi sürecinin müzik endüstrisi üzerindeki etkilerini inceledik. Bu bağlamda, Dudu, Ögüt ve Denizci'nin 2022 tarihli "Türkiye'de Müzik Emeğinin Durumu" başlıklı araştırması, pandeminin Türkiye'deki müzik emeği üzerindeki olumsuz etkilerini ortaya koymuştur. Dudu, Ögüt ve Denizci'nin araştırması, Türkiye'deki müzik emeği üzerindeki pandemi etkilerini detaylı bir şekilde irdelemiştir. Araştırmaya göre, müzisyenlerin güvencesiz çalışma koşulları ve gelir kayıpları, pandeminin getirdiği kısıtlamaların en çok etkilediği alanlardan biridir. Canlı performansların ve konser turnelerinin iptali, müzik endüstrisindeki yaratıcı emek gücünü olumsuz yönde etkilemiştir. Türkiye'de müzik endüstrisinin, pandemi öncesinde başlayan platformlaşma olgusundan etkilenmeye başladığı vurgulanmıştır. Ancak, pandemi süreciyle birlikte platformlaşmanın endüstri içindeki rolü daha da önem kazanmıştır. Canlı müzik performanslarının ve konser turnelerinin kapatılması, müzisyenlerin gelir kaynaklarını önemli ölçüde azaltmıştır. Araştırma, kısmi kapanma sürecinde en çok etkilenen alanları canlı müzik performanslarının yapıldığı sahneler, müzisyenler, tiyatro sanatçıları, tiyatro salonları, sinema salonları ve film festivalleri olarak belirlemiştir. Müzik endüstrisindeki yaratıcı emek gücü, sahnelerin ve konser turnelerinin kapatılmasıyla birlikte önemli ölçüde olumsuz etkilenmiştir. Dünya genelinde olduğu gibi Türkiye'de de müzisyenlerin gelirlerinin büyük bir kısmını canlı performanslar, konserler ve turnelerden elde ettiği bilinmektedir. Araştırma, müzisyenlerin doğrudan dinleyicilere ulaşabilecekleri, canlı müzik yapabilecekleri bir platformun eksikliğine vurgu yapmaktadır.

Bu araştırma, pandeminin müzik endüstrisindeki yaratıcı emek gücü üzerindeki etkilerini anlamamıza önemli bir katkı sağlamaktadır. Pandemi sürecinde müzik endüstrisindeki aktörler arasındaki güçlükleri anlamak, gelecekte benzer durumlar için daha etkili destek mekanizmaları oluşturulmasına yönelik önemli ipuçları sunmaktadır. Özellikle, müzisyenlerin gelir kaynaklarını çeşitlendirebilecekleri ve doğrudan dinleyicilere ulaşabilecekleri platformların geliştirilmesi önerilmektedir.

1.2. Amaçlar

1.2.1. Sanatçıların İlerlemesini Hızlandırmak

- Projemizin temel amacı, henüz geniş kitleler tarafından keşfedilmemiş sanatçıların daha fazla dinleyiciye ulaşmasını sağlamaktır. Bu platform, sanatçıların yeteneklerini sergileyebilecekleri, interaktif bir ortam sunar.

1.2.2. Topluluğa Hızlı Erişim:

- Sanatçılar, projemiz aracılığıyla hızlı bir şekilde kendi topluluklarını oluşturabilir ve dinleyicileriyle etkileşimde bulunabilirler. Yorumlar ve geribildirimler, sanatçıların kendilerini daha hızlı geliştirmelerine olanak sağlar.

1.2.3. Destek ve Bağış İmkanları:

- Dinleyiciler, sanatçılara destek olmak için kripto paraları kullanarak bağış ve aboneliklerde bulunabilirler. Bu destekler, sanatçıların müzik ekipmanlarını geliştirmeleri ve daha kaliteli içerik üretmeleri için kullanılabilir.

1.2.4. Kripto Para ile Hızlı Ödemeler:

- Projemiz, kripto paraları ödeme yöntemi olarak kullanarak, geleneksel para hizmetlerine göre daha ucuz ve hızlı ödeme seçenekleri sunar. 7/24 gerçekleştirilebilen ödemeler, aracı firmaların kesintilerini ortadan kaldırarak sanatçıya daha fazla gelir sağlar.

1.2.5. Takip Edilebilir İşlemler:

Ödeme işlemleri, şeffaf ve güvenli bir şekilde takip edilebilir olacaktır. Bu, sanatçıların destekleri ve bağışları kolayca izleyebilmelerini sağlar.

1.2.6. Küresel Duyuru ve Tanıtım:

Sanatçılar, kendi müziklerini dünyaya duyurmak için projemizi kullanabilirler. Platform, sanatçılara kendi kitlelerine ulaşma ve geniş kitlelere seslerini duyurma fırsatı sunar.

1.2.7. Ödeme Yöntemi ve Güvenlik:

- Kripto paralar, hızlı ve güvenli ödemeleri mümkün kılar. Proje, bu ödemeleri gerçekleştirmek ve takip etmek için gerekli güvenlik önlemlerini alarak sanatçılara ve dinleyicilere güvenli bir platform sunar.

Bu proje, müzik dünyasında yer almak isteyen sanatçılara daha geniş bir görünürlük ve destek sağlayarak onların gelişimini hızlandırmayı amaçlamaktadır. Aynı zamanda dinleyicilere, sevdikleri sanatçılara daha yakından destek olma ve etkileşimde bulunma şansı sunar. Projemiz, kripto paraların hızlı ve düşük maliyetli ödemeleriyle daha verimli bir şekilde çalışırken, şeffaf işlemlerle güvenliği de sağlar.

2. Kullanılacak Teknolojiler

Proje tasarlanırken aşağıdaki teknolojilerden yararlanılması düşünülüyor geliştirme aşamasında bunlar farklılık gösterebilir ve daha fazla teknolojiden yararlanılabilir.

2.1 Web Geliştirme:

- Frontend (Ön Yüz): HTML, CSS, JavaScript (React)
- Backend (Arka Yüz): Node.js (Express.js gibi), Python, Java (Spring Boot)

2.2 Canlı Yayın Protokolleri:

- WebRTC (Web Real-Time Communication)
- RTMP (Real-Time Messaging Protocol)
- HLS (HTTP Live Streaming)

2.3 Veri tabanı:

- PostgreSQL

2.4 Sunucu ve İstemci İletişimi:

- WebSocket teknolojisi kullanılabilir. Socket.io gibi kütüphanelerle entegrasyon yapılabilir.

2.5 Güvenlik ve Kimlik Doğrulama:

- OAuth 2.0 protokolü ile sosyal medya hesapları üzerinden giriş
- JWT (JSON Web Tokens) gibi kimlik doğrulama mekanizmaları

2.6 Ödeme Yöntemi:

- Solidity, Metamask

2.7 Mobil Uygulama:

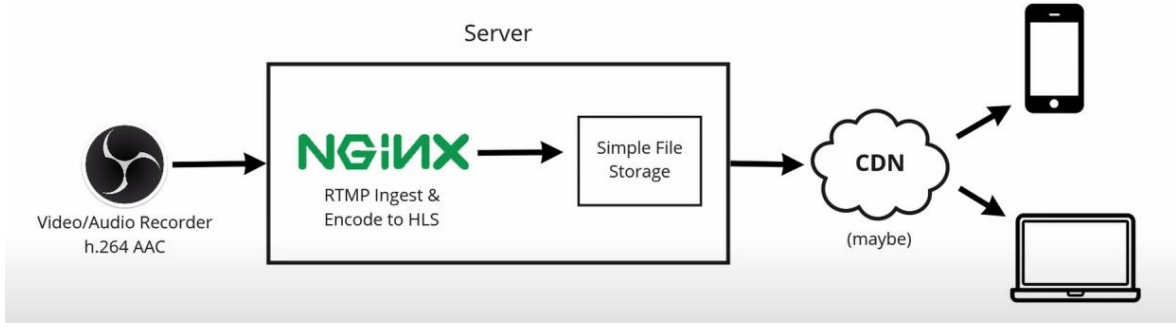
- React Native

Projenin ilk aşamasında, NGINX kullanarak RTMP sunucusu kurulumu gerçekleştirilecektir. NGINX'ın indirilip kurulmasının ardından, RTMP modülü konfigürasyon dosyasına eklenerek sunucu başlatılacaktır. Bu temel adım, canlı yayınların etkili bir şekilde yönetilebilmesi için gerekli altyapıyı oluşturacaktır. Daha sonra, Node.js ve Express kullanılarak kullanıcı doğrulama sistemini içeren bir API geliştirilecektir. Bu API, kullanıcıların kayıt olmalarını, giriş yapmalarını ve özel RTMP anahtarları oluşturmalarını sağlayacaktır. Kullanıcı doğrulama sistemi, başarılı girişlerden sonra kullanıcılara özel JWT (JSON Web Token) sağlayarak güvenli bir kimlik doğrulama sürecini destekleyecektir.



Şekil 1. RTMP sunucusunun çalışma mantığı

Ardından, RTMP akışları ffmpeg kullanılarak HLS (HTTP Live Streaming) formatına dönüştürülecek ve NGINX ile bu HLS dosyaları servis edilecektir. Bu sayede, farklı cihazlardan ve tarayıcılardan erişilebilen bir yayın alt yapısı oluşturularak geniş bir kullanıcı kitlesine ulaşılabilecektir.



Şekil 2. RTMP sunucusunun CDN ile yayınlanma mantığı

Server kısmı bittikten sonra projenin web ve mobil uygulama kısmına geçilecek ve backend geliştirmesi için Spring Boot, frontend geliştirmesi için ise React kullanılacaktır. Kullanıcı arayüzünde kayıt, giriş, canlı yayınları görüntüleme ve kripto para ile bağış yapma gibi işlevler bulunacaktır. Bu adımlar, projenin kapsamlı bir yapıya sahip olmasını sağlayacak ve kullanıcıların canlı yayınlara hızlı ve etkili bir şekilde erişmelerini mümkün kılacaktır.

3. Proje Oluşturma

Proje gereksinimlerinizi belirleyip, örneğin, Java sürümü, Spring Boot versiyonu ve bağımlılıkları belirleyip projeyi oluşturacağız.

3.1. Veri tabanı ve JPA Entegrasyonu

3.1.1. Veri tabanı Seçimi ve Bağlantı Kurma:

Kullanılacak veri tabanlarını belirleyeceğiz (PostgreSQL). application.properties dosyasında gerekli veritabanı bağlantı bilgilerini yapılandıracağız.

3.1.2. JPA Entegrasyonu ve Entity Oluşturma:

JPA (Java Persistence API) kullanarak veri tabanı işlemleri gerçekleştirmek için entity sınıflarını oluşturacağız. İlişkileri ve tablo yapılarını entity sınıfları üzerinden tanımlayarak veri tabanı modelimizi oluşturacağız.

3.1.3. Neden PostgreSQL? :

- Açık Kaynak ve Ücretsiz: PostgreSQL, açık kaynak bir projedir ve ücretsiz olarak kullanılabilir. Bu, kullanıcıların yazılımı inceleme, değiştirme ve dağıtma özgürlüğüne sahip olduğu anlamına gelir.

- **Geniş Veri Tipi Desteği:** PostgreSQL, standart SQL veri tiplerine ek olarak, kullanıcı tanımlı veri tiplerini destekler. Ayrıca, JSON, XML, Hstore gibi modern veri tiplerine de destek verir.
- **Gelişmiş Uzantı ve Eklenti Desteği:** PostgreSQL, kullanıcılara özelleştirilmiş işlevsellik eklemelerine izin veren bir eklenti mekanizması sunar. Bu, özel işlevlerin veya özelliklerin eklenmesini kolaylaştırır.
- **ACID Uyumlu:** PostgreSQL, ACID (Atomicity, Consistency, Isolation, Durability) özelliklerini destekleyen bir veritabanıdır. Bu, veritabanının güvenilirliğini ve veri bütünlüğünü sağlar.
- **Gelişmiş Performans ve Optimizasyon:** PostgreSQL, sorgu optimizasyonu ve performans artırıcı özelliklere odaklanan gelişmiş bir sorgu planlama motoruna sahiptir. Bu, büyük veri setleri üzerinde etkili bir şekilde çalışmayı sağlar.
- **Çok Kullanıcılı Desteği:** PostgreSQL, eş zamanlı çok kullanıcılı ortamlarda güvenli bir şekilde çalışabilir. Kilitlenme ve eş zamanlılık kontrolü, aynı anda birçok kullanıcının veritabanına erişimini sağlar.
- **Geniş Topluluk Desteği:** PostgreSQL, büyük ve aktif bir açık kaynak topluluğa sahiptir. Bu topluluk, belgelendirme, sorun giderme ve güncellemeler konusunda destek sağlar.
- **Geniş Platform Desteği:** PostgreSQL, çeşitli işletim sistemleri (Linux, Windows, macOS) üzerinde çalışabilir. Ayrıca, çeşitli programlama dilleri ve araçlarla entegre edilebilir.
- **Yedekleme ve Kurtarma:** PostgreSQL, veritabanı yedekleme ve geri yükleme işlemleri için güçlü araçlara sahiptir. Bu, veri kaybını önlemek ve sistem çöküşlerine karşı güvenliği artırmak için önemlidir.
- **Güvenlik:** PostgreSQL, güvenlik konusunda ciddi bir odaklanmaya sahiptir. Gelişmiş yetkilendirme ve kimlik doğrulama mekanizmaları sunar.

Server kısmı bittikten sonra projenin web ve mobil uygulama kısmına geçilecek ve backend geliřtirmesi için Spring Boot, frontend geliřtirmesi için ise React kullanılacaktır. Kullanıcı arayüzünde kayıt, giriř, canlı yayınları görüntüleme ve kripto para ile bağıř yapma gibi işlevler bulunacaktır. Bu adımlar, projenin kapsamlı bir yapıya sahip olmasını sağlayacak ve kullanıcıların canlı yayınlara hızlı ve etkili bir şekilde erişmelerini mümkün kılacaktır.

3.1.4. Neden Spring Boot ? :

- Hız ve Kolaylık: Hızlı geliřtirme ve sıfır XML konfigürasyonu sayesinde projelerinizi hızlı bir şekilde başlatabilirsiniz.
- Modüler ve Geniřletilebilir: Kolay entegrasyon ve modüler yapısıyla ihtiyalarınıza uygun bir şekilde genişletilebilir.
- Standart Konfigürasyon: Varsayılan konfigürasyonlar ve standart proje yapıları sayesinde daha az kod yazma ihtiyacı ve hızlı başlangı.
- Spring Ekosistemi: Geniř Spring ekosistemi ile entegrasyon, çeřitli araç ve kütüphanelere kolay erişim sağlar.
- Gömülü Sunucu Desteğı: Uygulamalarınızı çeřitli gömülü sunucular üzerinde çalıştırabilir, bağımsızlık ve dağıtılabilirlik avantajları elde edebilirsiniz.
- Starter Projeler: Hazır projelerle belirli teknoloji yığınlarına sahip uygulamaları hızlıca başlatabilirsiniz.
- Spring Boot Actuator: Geliřmiş operasyonel özelliklerle uygulamanızı yönetme ve izleme imkanı sağlar.
- Bağımlılık Yönetimi: Maven veya Gradle gibi araçlarla kolayca bağımlılıkları yönetebilirsiniz.
- Mikro Hizmetlere Uyum: Mikro hizmet mimarisini destekler, modüler yapı ile büyük sistemleri kolayca oluřturmanızı sağlar.
- Büyük Topluluk Desteğı: Aktif bir topluluk, zengin belgeler ve kaynaklar ile desteklenir.

Sonuç olarak hızlı geliştirme, modüler yapı, geniş entegrasyon olanakları ve büyük bir topluluk desteği ile Java tabanlı projeler için güçlü bir tercih olduğu için projemizin backend işlemlerinde Spring Boot kullanacağız.

3.2.Controller ve Servis Katmanlarının Oluşturulması

3.2.1. Controller Oluşturma:

HTTP isteklerini karşılamak üzere controller sınıfları oluşturup. `@RestController` anotasyonlarını kullanarak HTTP endpoint'lerini tanımlayacağız.Bu katman ile kullanıcı ile projemiz arasındaki ilişkiyi kurmuş olacağız.

3.2.2 Service Oluşturma:

İş mantığı işlemlerini yürüten servis sınıflarını oluşturup Servis sınıfları, controller ve repository sınıfları arasında bir arayüz sağlamak üzere tasarlayacağız.

3.3. React.JavaScript Front-end Kütüphanesi Kullanımı

Dinamik HTML sayfalarını oluşturup tasarımını yapacağız. Controller sınıfları ile entegre edilecek şekilde dinamik içerikleri HTML sayfalarına ekleyeceğiz.

Front-end geliştirmek için JavaScript kütüphanesi olan React'ı kullanacağız. Client-side kodlarınızı yazarak RESTful servislerle iletişim kuracağız.

3.3.1. Neden React ? :

React, bileşen tabanlı yapısı, hızlı Virtual DOM kullanımı, geniş topluluk desteği, React Native ile mobil uygulama geliştirmeye olanak tanınması, kolay öğrenilebilirliği, bağımsız kullanım imkanı, güçlü araç desteği, güncel ve sürekli geliştirilen yapısı gibi özellikleriyle öne çıkar. Bu özellikler, React'in sade, performanslı, ve yaygın kullanılan bir kullanıcı arayüzü kütüphanesi olmasını sağlar. Bu nedenlerden dolayı react.js'i projemizde tercih edeceğiz.

3.4. Güvenlik ve Kimlik Doğrulama

3.4.1. Spring Security Entegrasyonu:

Spring Security kullanarak kimlik doğrulama ve yetkilendirme işlemlerini gerçekleştireceğiz. WebSecurityConfigurerAdapter sınıfını genişleterek güvenlik ayarlarını yapacağız.

3.5.Uygulama Testleri ve Hata Ayıklama

3.5.1. JUnit veya TestNG Kullanma:

Birim testleri yazarak uygulamanızı test edip. Spring Boot'un desteklediği test çerçevelerini (JUnit, TestNG) kullanarak test senaryolarınızı oluşturacağız.

3.5.2. Logging ve Hata Ayıklama:

Log mekanizmalarını (örneğin, SLF4J) kullanarak uygulamanızın durumunu kayıt altına alıp kontrol edebileceğiz.

3.6.Kripto Para ile Bağış Sistemi:

- Pandemi Sonrası Artan Çevrimiçi Bağışlar: 2020'deki Covid-19 pandemisi, dünya genelinde çevrimiçi bağışlarda önemli bir artışa neden oldu. Sivil toplum kuruluşları (STK) ve hayır kurumları, geleneksel ödeme yöntemlerinin yanı sıra kripto paraları da kabul etmeye başladı.
- UNICEF ve Greenpeace Örneği: Önde gelen kuruluşlar, örneğin UNICEF ve Greenpeace gibi, kripto paraları bağış kabul etmeye başlayarak bu alandaki değişime öncülük ettiler.
- Blokzinciri ve Şeffaflık: Blokzinciri teknolojisi, yapılan bağışların şeffaf, sınır ötesi ve düşük maliyetli bir şekilde takip edilmesini sağlar. Blokzinciri, herkesin belirli bir bağışın hangi blokta ve hangi veriyi içerdiğini görebileceği şeffaf bir yapı sunar.

3.6.1. Kripto Para ile Ödeme Yöntemi ve MetaMask Entegrasyonu:

- MetaMask Eklentisi: Kullanıcılara sitemize kayıt olduktan sonra MetaMask eklentisi ile bir kripto para cüzdanı oluşturulur. MetaMask, kullanıcılara Ethereum tabanlı cüzdanlarını yönetme yeteneği sağlar.
- Cüzdanın EVM Adresi (IBAN): Her kullanıcının cüzdanına, Ethereum sanal cüzdan adresi veya EVM (Ethereum Virtual Machine) adresi atanır. Bu adres, kullanıcının cüzdanına para göndermek isteyenlerin kullanması için bir IBAN gibi hizmet eder.
- Para Yatırma ve Bağış: Kullanıcılar, kendi cüzdanlarına para yatırabilirler. Daha sonra bu parayı istedikleri yayıncıya bağış yapmak için kullanabilirler. Blokzinciri sayesinde, bu bağış işlemleri şeffaf bir şekilde kaydedilir ve takip edilebilir.
- Aracısız İşlemler: Kripto paraların kullanılması, araçlara ihtiyaç duyulmadan doğrudan kullanıcıdan yayıncıya para transferi yapılabilmesine olanak tanır. Bu durum, bağış işlemlerinin hızlı ve düşük maliyetli olmasını sağlar.

3.6.2. Güvenlik ve İşlemler Arası Kesintisizlik:

- Güvenlik Önlemleri: MetaMask ve blokzinciri teknolojisi, kullanıcıların cüzdanlarını güvenli bir şekilde yönetmelerine olanak tanır. Kripto para işlemleri, güvenlik açısından yüksek standartlara sahiptir.
- İşlemler Arası Kesintisizlik: Kripto paraların merkezi olmayan doğası, işlemler arası kesintisizliği sağlar. Herhangi bir aracı olmaksızın, kullanıcıdan kullanıcıya hızlı ve sorunsuz para transferleri gerçekleşir.

3.7. React Native

Projemizin mobil uygulama kısmını geliştirmek için React Native kullanacağız. React Native, JavaScript tabanlı bir çerçeve olup, hem iOS hem de Android platformları için tek bir kod tabanında mobil uygulama geliştirmeyi sağlar.

3.7.1. apraz Platform Desteęi:

React Native, apraz platform geliřtirmeye odaklanır, bu da geliřtirilen uygulamanın hem iOS hem de Android platformlarında alıřabilmesi anlamına gelir. Bu zellik, geliřtirme srecini hızlandırır ve maliyetleri azaltır.

3.7.2. Tek Bir Kod Tabanı:

React Native, aynı JavaScript kodunu kullanarak hem iOS hem de Android uygulamalarını oluřturmanıza olanak tanır. Bu, geliřtirme srecini daha tutarlı ve etkili hale getirir.

3.7.3. Hızlı Geliřtirme ve Canlı Yeniden Ykleme:

React Native, canlı yeniden ykleme zellięi sayesinde geliřtiricilere anlık deęiřiklikleri grme ve test etme imkanı sunar. Bu, geliřtirme srecini hızlandırır ve anında geri bildirim almanızı saęlar.

3.7.4. Modler ve Geniřletilebilir Yapı:

React Native, bileřen tabanlı bir yapıya sahiptir ve modler tasarımı sayesinde uygulamanızı iřteęe baęlı olarak geniřletebilirsiniz. Bu, geliřtirme srecini daha esnek hale getirir.

3.7.5. nc Taraf Ktphanelerin Kolay Entegrasyonu:

React Native, birok nc taraf ktphaneyi kolayca entegre etme yeteneęi sunar. Bu, projenize ek zellikler eklemenizi saęlar ve geliřtirme srecini optimize eder.

3.8. Solidity Nedir?

Solidity, Ethereum blok zinciri üzerinde akıllı kontratlar oluşturmak için kullanılan bir programlama dilidir. Ethereum, merkezi olmayan uygulamalar (DApps) ve akıllı kontratlar gibi inovatif uygulamaların geliştirilmesini sağlayan bir blok zinciri platformudur. Solidity, Ethereum'un akıllı kontratları çalıştırma sanatını öğrenen ve kullanıcıların kendi merkezi olmayan uygulamalarını (DApps) oluşturmaya imkan tanıyan bir dil olarak öne çıkmaktadır.

Akıllı Kontratlar İçin Tasarlandı: Solidity, Ethereum blok zinciri üzerinde çalışan akıllı kontratlar için tasarlanmıştır. Akıllı kontratlar, belirli bir koşul veya işlem gerçekleştiğinde otomatik olarak çalışan, kod içeren sözleşmelerdir. **Statik ve Güvenli:** Solidity, statik tip sistemine sahip bir dildir ve kodun güvenli olmasını sağlamak için tasarlanmıştır. Bu, programcıların hataları önceden fark etmelerine ve düzeltmelerine yardımcı olabilir. **Ethereum Virtual Machine (EVM) İçin Derlenir:** Solidity ile yazılan akıllı kontratlar, Ethereum Virtual Machine (EVM) adlı sanal bir makine üzerinde çalıştırılır. Bu, herhangi bir Ethereum düğümü tarafından yürütülebilen bir standart sağlar.

Ethereum Topluluğu Tarafından Desteklenir: Solidity, Ethereum topluluğu tarafından geliştirilmekte ve desteklenmektedir. Bu, geliştiricilerin sorularını sormak, kaynaklara erişmek ve güncellemeleri takip etmek için zengin bir kaynak havuzuna sahip oldukları anlamına gelir.

Sözleşme-Oriented: Solidity, sözleşme yönelimli bir dil olarak tasarlanmıştır. Bu, programcıların akıllı kontratları ve sözleşmeleri düşünerek tasarımlarını sağlar.

Özetle, Solidity, Ethereum blok zinciri üzerinde akıllı kontratlar oluşturmak için kullanılan bir programlama dilidir. Ethereum'un merkezi olmayan uygulamalarını (DApps) geliştirmek ve blok zinciri tabanlı projeleri inşa etmek isteyen geliştiriciler tarafından yaygın olarak kullanılmaktadır.

Bu detaylar, kripto para ile bağış sisteminizin nasıl işlediğini ve kullanıcılar arasındaki bağış işlemlerini nasıl kolaylaştırdığınızı açıklar. Bu sistemin avantajları, geleneksel ödeme yöntemlerine göre daha hızlı, düşük maliyetli ve şeffaf bir deneyim sunmaktadır.

4. YAPILAN ÇALIŞMALAR

4.1. Planlama

Covid-19 zamanı oluşan problemleri belirlemek ve ortadan kaldırmak için en doğru yolu belirlemek adına çeşitli çalışmalarda ve bir çok araştırmalarda bulunduk. Bu araştırmalara göre, Covid-19 zamanında en olumsuz şekilde etkilenen gruplardan biri de şüphesiz müzik sektöründe faaliyet gösteren kişiler olmuştur(Özarıslan,2021). Kreksa'nın ‘‘Pandemi Sanatçı ve Kùltür Profesyonellerini Ekonomik Olarak Nasıl Etkiledi?’’ yapmıř olduđu araştırma raporunda ise müzisyenlerin %78'nin temel geçim kaynağı %67'nin ise tek gelir kaynağı pandemi döneminde tamamen durmuştur(KREKSE,2020).

4.2. Analiz

Bu araştırmalar sonucunda elde ettiğimiz çıkarımlar; Covid-19 gibi pandemilerde etkilenen sektörlerin daha az etkilenmesini sağlamak adına müzisyenlerin ek gelir kaynağı olarak canlı yayınlar ve dijital konserler düzenleyebileceğı bir platform geliştirme kararı aldık.

4.3. İhtiyaç Analizinin Yapılması

4.3.1. Kullanım Kolaylığı

Uygulama ve Web sitesi herhangi bir kullanıcı tarafından kolaylıkla kullanılabilir ve yalın tasarıma sahip olacaktır. Bağış, abonelik işlemleri gibi yorucu işlemler dahi herkesin anlayacağı biçimde olacaktır.

4.3.2. Güvenilirlik

Kullanıcıların bağışları, hesapları metamask ile korunacak ve gerekli kriptoloji işlemleri ile kişisel bilgileri korunacaktır.

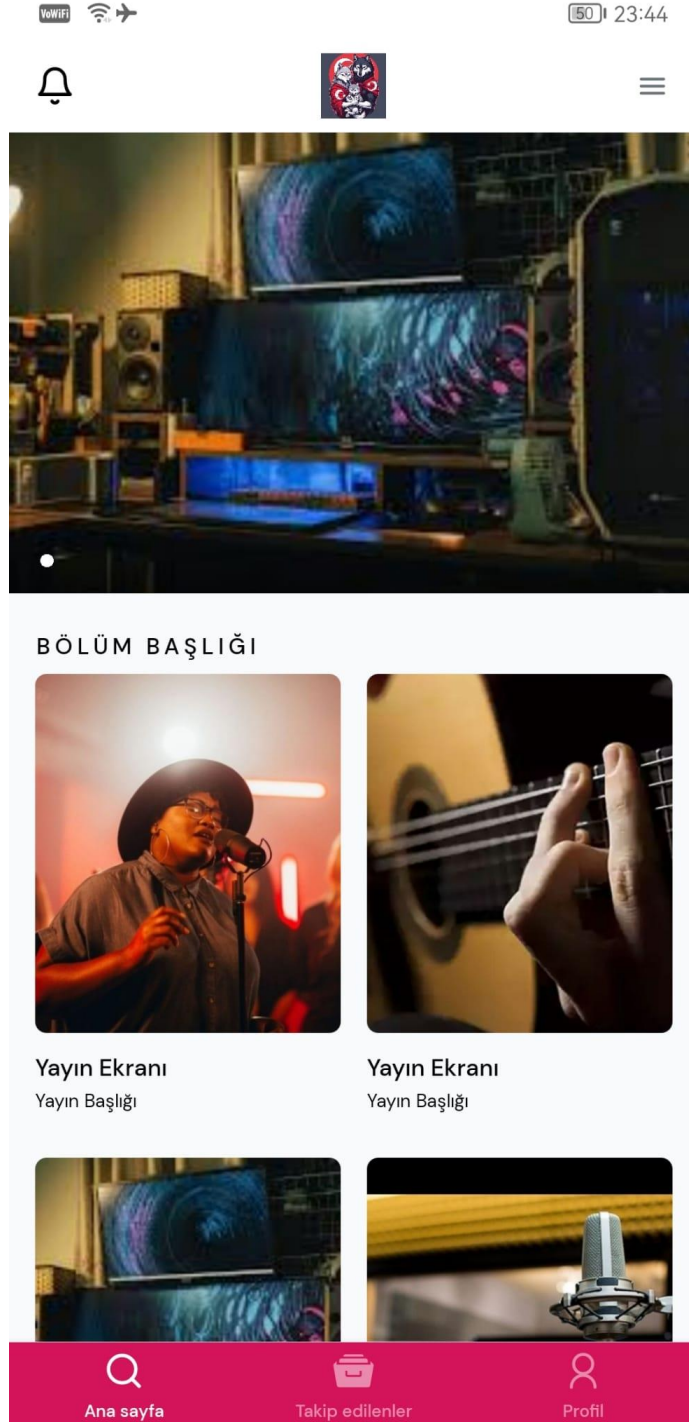
4.3.3. Performans

Kullanıcıların kullanırken sunucularda çökme, donma gibi hatalarla karşılaşmaması için sunucuların yapısı ve kullanılacağı alan özenle belirlenecektir.

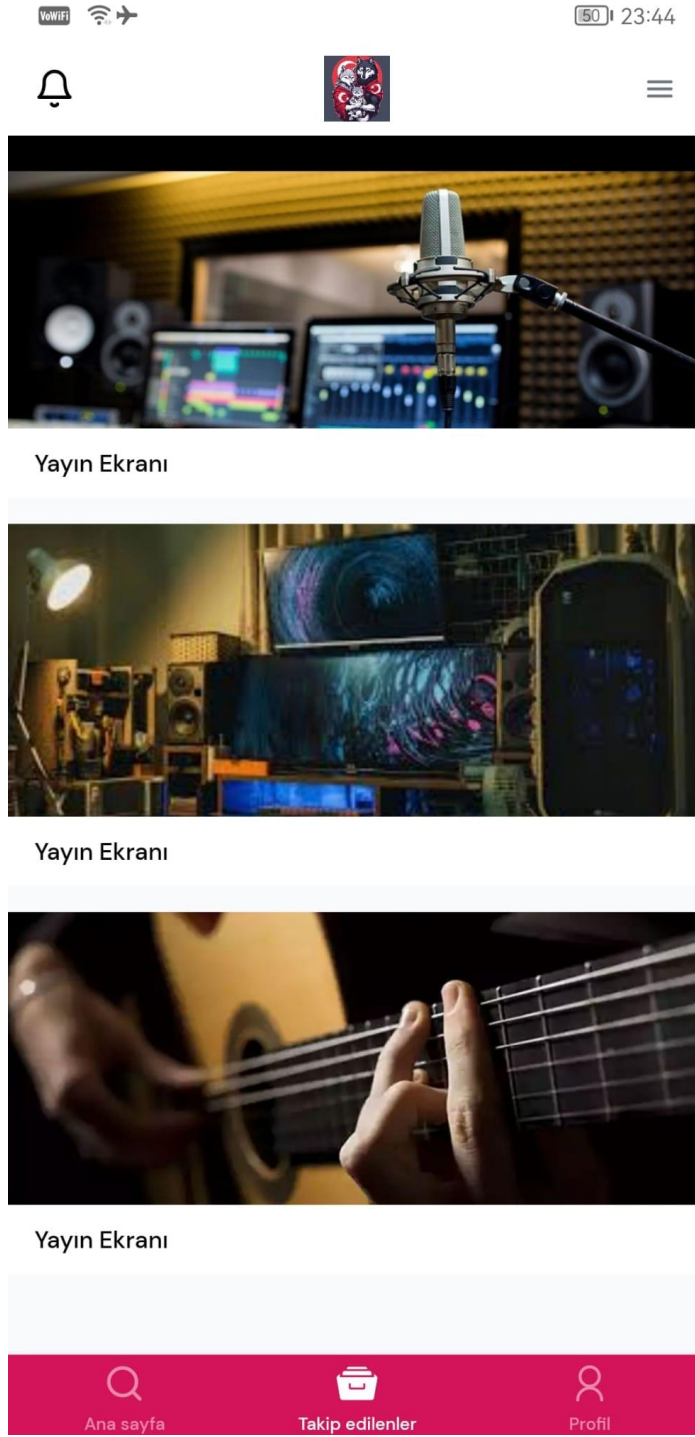
4.3.4. Desteklenebilirlik

Mobil ve Web'den kolaylıkla erişim sağlanabilecektir.

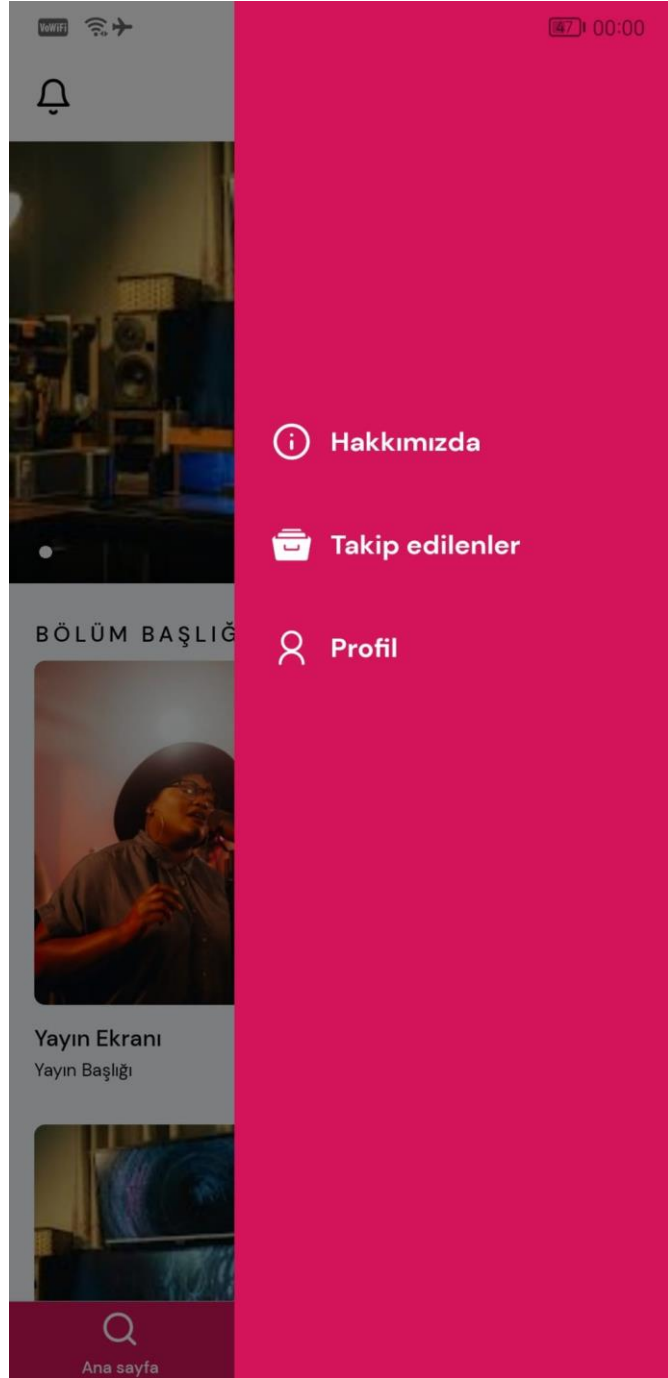
5. PROJE TASARIMI



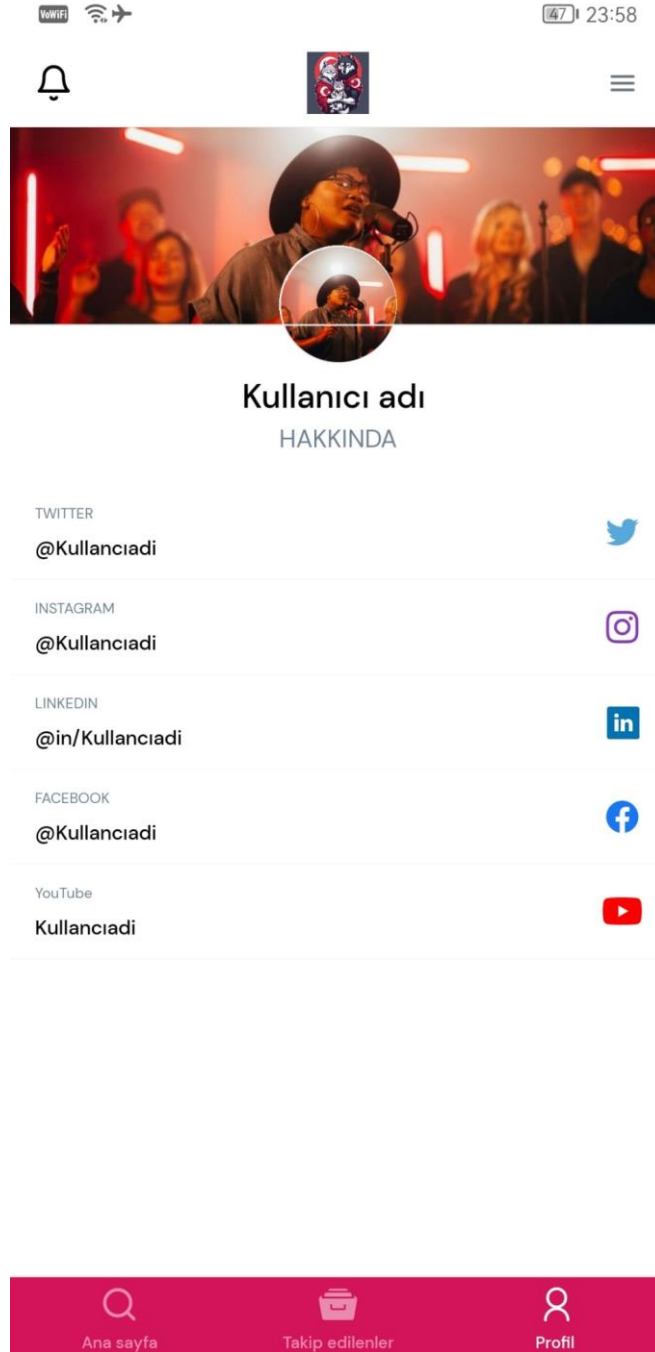
Şekil 3. Ana sayfa



Şekil 4. Takip Edilenler



Şekil 5. Sidebar



Şekil 6. Kullanıcı Profili

6. KAYNAKLAR

1. Medium. (tarih yok). 2023 tarihinde Medium Website: <https://medium.com/kodcular/react-native-nedir-7b333d319597> adresinden alındı
2. Medium. (tarih yok). 2023 tarihinde Medium website: <https://medium.com/turkcell/postgresql-nedir-356b6042a88a> adresinden alındı
3. MetaMask. (tarih yok). 2023 tarihinde MeteMask Web Site: <https://support.metamask.io/hc/tr/articles/360015489531-MetaMask-a-giri%C5%9F> adresinden alındı
4. Nedir. (tarih yok). 2023 tarihinde Nedir.com: <https://www.nedir.com/rtmp> adresinden alındı
5. NodeJs. (tarih yok). 2023 tarihinde NodeJs Website: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> adresinden alındı
6. Spring. (tarih yok). 2023 tarihinde Spring WebSite: <https://spring.io/guides/gs/spring-boot/> adresinden alındı
7. Wikipedia. (tarih yok). 2023 tarihinde Wikipedia Website: <https://en.wikipedia.org/wiki/Solidity> adresinden alındı
8. Özarslan, Z. (2021). Yaratıcı ve Kültürel Endüstriler ve Covid-19 Pandemi Döneminde Türkiye'de Kültür Ve Sanat Sektörlerinin Durumu. Alternatif Politika, 371-408.
9. KREKSA. (2020). Pandeminin Bağımsız Sanatçıların Ekonomik Durumuna Etkileri. KREKSA Kültür.

STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Projenin benzeri birçok platform vardır ancak kullanım alanları ve kullanıcı tipleri açısından müzisyenler tarafından pek uygun olmamakla birlikte ülkemizde bir gelir kapısı olacak yeterli bir getiri sağlamamaktadır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Web Tasarımı ve Programlama – Html, Css
Mobil Uygulama Geliştirme
Yazılım Gereksinimi Mühendisliği
Nesne Yönelimli Programlama

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

b) Çevre sorunları:

c) Sürdürülebilirlik:

d) Üretilebilirlik:

e) Etik:

f) Sağlık:

g) Güvenlik:

h) Sosyal ve politik sorunlar:

| | |
|-----------------|--|
| Ad Soyad | |
| Tel no | |
| E-posta | |
| Adres | |

Özgeçmiş

Özgeçmiş Bilgileri :