**ANKARA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

****

**BLM 4062 PROJE RAPORU**

**Tiyatro Duyuru Uygulaması**

**Betül Öğmen**

**19290263**

**Danışman : Yılmaz AR**

**11 Haziran 2024**

# İÇİNDEKİLER

[1. RAPORUN BÖLÜMLERİ 1](#_Toc39681033)

[1.1. Özet 1](#_Toc39681034)

[1.2. Giriş 2](#_Toc39681036)

[1.3. Gelişme 3](#_Toc39681037)

[1.4. Sonuç 15](#_Toc39681037)

# RAPORUN BÖLÜMLERİ

## Özet

Bu projede, Flutter teknolojisi kullanılarak tiyatro koltuğu seçme uygulaması geliştirilmiştir. Uygulama, kullanıcıların mevcut tiyatroları görüntülemesine, koltuk seçmesine ve bilet rezervasyonu yapmasına olanak tanır. Bu projede, kullanıcı yetkilendirmesi, giriş ekranı, tiyatrolar ekranı, bilet ekranı, koltuk seçme ekranı ve tarihi geçen etkinliklerin yönetimi gibi çeşitli bölümler bulunmaktadır. Uygulamada iki tip kullanıcı rolü vardır: yönetici (admin) ve kullanıcı (user). Yönetici, yeni etkinlikler ekleyebilir, mevcut etkinlikleri silebilir ve tüm etkinlikleri görüntüleyebilirken, kullanıcılar sadece etkinlikleri görüntüleyebilir ve bilet satın alabilir. Bu yapı, uygulamanın güvenliğini sağlamak için önemlidir.

Giriş ekranında, kullanıcılardan e-posta adresi ve şifre talep edilir. Girilen bilgiler Supabase'in kimlik doğrulama API'sine gönderilir ve başarılı bir giriş durumunda oturum tokeni döndürülür. Bu token, uygulamada kullanıcının oturum açık kaldığını gösterir. Ayrıca, uygulamada kayıtlı kullanıcıların bilgileri Supabase'in 'users' tablosunda saklanır. Tiyatrolar ekranında, Supabase veritabanından çekilen etkinlikler listelenir. Etkinlikler 'EventWidget' sınıfı aracılığıyla gösterilir ve her etkinlik için etkinlik adı, türü, yapımcısı, tarihi ve resim içeren bir kart oluşturulur. Yeni etkinlik ekleyebilme yetkisi yalnızca yönetici kullanıcılara tanınır. Etkinliklerin görüntülenmesi ve yönetilmesi, Supabase'in veri yönetimi özelliklerini kullanarak gerçekleştirilir.

Bilet ekranında, tiyatrolar sayfasında seçilen bir tiyatronun üzerine tıklanınca bilet ekranı görünür. Bilet ekranında, tiyatronun adı, yeri, kayıtlı resmi, yapımcısı gibi bilgiler tiyatro ekranından çekilir ve ekranda bilet kodu oluşturulur. 'Koltuk Seç' butonu ile koltuk ekranına gidilirken, 'Bilet Al' ekranı ile bütün değişkenler tamamlanır ve biletin son hali görünür. Koltuk ekranına geçilmeden önce bir pop-up ekranında etkinliğin hangi salondan alınacağı seçilir. Bilet ekranında pop-up ile seçilmiş olan salon numarası doğrulanır ve doğru salondan bilet alınması sağlanır.

Koltuk seçme ekranında tiyatro salonunun dijital versiyonu görünür. Eğer daha önceden alınmış koltuklar var ise, bunlar salonun veri tabanından kontrol edilir. Dolu koltukların rengi ekranda kırmızı, boş koltukların rengi ise yeşil olarak gösterilir. Koltukların numaraları gridlerin üzerinde tanımlıdır ve bu gridlerin üzerine tıklandığında yer ayırılacak koltuklar seçilmiş olur. Tekrar basıldığında koltuklar seçimden kaldırılabilir ve hangi koltuğun seçili olduğu ekranın alt kısmında görünür. 'Yer Ayırt' butonuna basıldığında ise, seçilen koltuk numaraları salonun veri tabanında dolu olarak işaretlenir. Bu sayede o koltuğa o anda başka bir kullanıcı bilet ayıramaz ve koltuk zaten dolu uyarısı alınır.

Bilet Al ekranında, seçilen etkinliğin adı, tarihi, salon numarası, koltuk numaraları ve bilet kodu gösterilir. Bu noktada bilet artık oluşmuştur ve veri tabanındaki 'tickets' tablosunda saklanır.

Proje, her biri belirli bir amaca hizmet eden birkaç Dart dosyasına ayrılmıştır. Bu dosyalar şunlardır: `ticket\_page.dart` bilet rezervasyon sayfasını yönetir, `theaters.dart` mevcut tiyatroları listeler, `ticket\_style.dart` biletlerin stilini tanımlar, `tickets.dart` biletler için veri modelini içerir, `seat.dart` koltukların temsili ve işlevselliğini yönetir, `addseat.dart` yeni koltukların eklenmesini sağlar ve `app\_color.dart` uygulamadaki renk temalarını tanımlar.

Ticket\_style.dart dosyası, uygulamada bilet bilgilerinin görüntülenmesi için kullanılan metin stillerinin tanımlanmasından sorumludur. Bu dosyada, biletlerde olması planlanan değişkenler theater.dart dosyasından aktarılır ve bu sayede ilgili verilere erişilir. Sanal bir bilet görüntüsü vermek amacıyla Flutter'ın TextStyle sınıfını kullanır ve çeşitli çizgi fonksiyonları ile bilet şekli çizilir. Sayfada biletin öncelikli ve eşsiz değişkeni olan bilet kodu oluşturulur.

Tickets.dart dosyası, uygulamadaki bilet konseptini modelleyen bir Ticket sınıfını tanımlar. Bu sınıf, 'tickets' tablosuna veri eklenebilmesi için oluşturulmuştur. Tickets tablosu, 'ticket\_id', 'ticket\_type', 'price', 'event\_id', 'ticket\_date', 'event\_name', 'venue', 'event\_image', 'user\_email', 'ticket\_no', 'reserved\_seats' ve 'salon' sütunlarından oluşur. 'ticket\_id' primary key, 'event\_id' ise events tablosu ile bağlantılıdır ve id değerleri events'tan gelir. 'saloonone' ve 'saloontwo' tablolarında birbirine bağlı üç primary key bulunur: 'event\_name', 'seat\_no', ve 'event\_date'. Bunun böyle tanımlanmasının sebebi, aynı koltuğa iki bilet kesilmesini önlemektir. Bu tablolarda ayrıca koltuğun dolu olup olmadığını tutan bir 'seat\_state' sütunu bulunur.

Seat.dart dosyasında, ekranda görünen salon koltukları tasarlanır. Şekillerin düzeni, rengi ve font çeşitleri belirlenir. Koltuklar önce uygulamanın arayüzünde oluşturulur ve sonradan veritabanı işlemleri devreye girer. 'saloonone' ve 'saloontwo' tablolarında daha önceden ayırtılmış koltuk bilgileri tutulur. Bu tablolardan hangi verilerin çekilmesi gerektiği, bilet sayfasından çekilen etkinliğin adı, tarihi ve salonu ile belirlenir. Bu tarihler ile çakışan satırlar var ise, o satırlardaki koltuk numaraları ekranda kırmızıya döner. Ticket\_style.dart dosyasında tanımlanmış bilet kodu, sayfa açılır açılmaz oluşturulur.

Giriş ekranında, kullanıcılardan e-posta adresi ve şifre talep edilir. Girilen bilgiler Supabase'in kimlik doğrulama API'sine gönderilir ve başarılı bir giriş durumunda oturum tokeni döndürülür. Bu token, uygulamada kullanıcının oturum açık kaldığını gösterir. Ayrıca, uygulamada kayıtlı kullanıcıların bilgileri Supabase'in 'users' tablosunda saklanır. Tiyatrolar ekranında, Supabase veritabanından çekilen etkinlikler listelenir ve etkinlikler 'EventWidget' sınıfı aracılığıyla gösterilir. Her etkinlik için etkinlik adı, türü, yapımcısı, tarihi ve resim içeren bir kart oluşturulur. Yeni etkinlik ekleyebilme yetkisi yalnızca yönetici kullanıcılara tanınır. Etkinliklerin görüntülenmesi ve yönetilmesi, Supabase'in veri yönetimi özelliklerini kullanarak gerçekleştirilir. Uygulama, sadece tarihi geçmemiş etkinlikleri listeler. Bu, SQL sorgularına eklenen bir koşulla sağlanır ve böylece geçmiş tarihli etkinlikler kullanıcı arayüzünde görünmez. Bu özellik, kullanıcılara güncel ve ilgili bilgilerin sunulmasını sağlar.

Bu metin, projenizin temel işleyişini, kullanılan teknolojileri ve uygulamanın çeşitli ekranlarını detaylı bir şekilde açıklamaktadır. Bu şekilde, raporunuzda hem projenizin genel yapısını hem de teknik detaylarını etkili bir şekilde özetlemiş olursunuz.

## Giriş

Tiyatro gösterilerinin düzenlendiği özel bir tiyatro salonu, etkinliklerini yayımladığı bir online mobile uygulamaya mutlaka ihtiyaç duyacaktır. Etkinliklerin yapılma tarihi, nerede yapılacağını online bir şekilde duyurması gerekecektir. Bu proje öyle bir amaca hizmet etmesi için düzenlenmiştir.

Günümüzde dijital dönüşüm, tüm sektörleri etkilediği gibi sanat dünyasını da etkilemektedir. Tiyatrolar, sanatsal ifadelerini ve etkileşimlerini genişletmek için teknolojiyi benimsemek zorundadır. Bir online mobil uygulama, tiyatroların geniş bir kitleye ulaşmasını sağlayarak sanatın daha erişilebilir olmasına katkıda bulunur. Kullanıcılar, cep telefonları veya tabletler aracılığıyla istedikleri yerden ve zamanda tiyatro etkinlikleri hakkında bilgi sahibi olabilirler. Bu, fiziksel sınırlamaları aşarak sanatseverlerin ve tiyatro meraklılarının tiyatroya olan erişimini genişletir.

Mobil uygulamalar, kullanıcı deneyimini önemli ölçüde iyileştirir. Etkinlik biletlerini online olarak satın almak, koltuk seçimi yapmak, etkinlik programlarını gözden geçirmek ve sanatçılar hakkında bilgi edinmek gibi işlemler kullanıcıların elinin altında olur. Ayrıca, uygulama aracılığıyla kullanıcılara özel indirimler, bildirimler ve güncellemeler sunulabilir, bu da kullanıcıların tiyatro ile olan bağını güçlendirir.

Tiyatrolar için güçlü bir pazarlama ve tanıtım aracıdır. Yeni etkinlikler, özel gösterimler ve sanatçı haberleri hızlı ve etkili bir şekilde duyurulabilir. Push bildirimleri, kullanıcıları yaklaşan etkinlikler ve özel teklifler hakkında bilgilendirir. Bu, etkinliklerin daha fazla insana ulaşmasını ve dolayısıyla daha yüksek bilet satışlarına yol açmasını sağlar.

Kağıt bilet kullanımını azaltarak çevre dostu bir alternatif sunar. Kullanıcılar, uygulama üzerinden biletlerini satın alabilir ve bunları dijital olarak saklayabilirler. Giriş sırasında, biletler QR kodlar veya benzeri dijital yöntemlerle kolayca taranabilir. Bu, hem kullanıcıların hem de tiyatro personelinin işlerini kolaylaştırır ve etkinlik giriş süreçlerini hızlandırır.

Dijital bir platform olarak, mobil uygulamalar, tiyatroların uzun vadeli sürdürülebilirliğini ve gelişimini destekler. Dijital ortamda varlık göstermek, özellikle genç nesiller arasında tiyatroya olan ilgiyi artırabilir ve gelecekteki sanatseverlerin ve destekçilerin yetişmesine katkıda bulunabilir. Ayrıca, dijital platformlar, pandemi gibi olağanüstü durumlarda fiziksel mekanların kapalı olması gibi durumlarda bile sanatsal etkinliklerin devam etmesine olanak tanır. Bu durum, tiyatroların zor zamanlarda bile sanatı sürdürme ve izleyicileriyle bağlarını koruma kapasitesini artırır. Tiyatroların finansal yönetimini ve kaynak optimizasyonunu iyileştirebilir. Online bilet satışları, gelir akışını düzenler ve finansal planlama için değerli veriler sunar. Ayrıca, dijital reklamlar ve sponsorluklar yoluyla ek gelir elde etme fırsatları da sağlar.

Online bir platform, yerel sınırları aşarak küresel bir izleyici kitlesine ulaşmayı mümkün kılar. Bu, uluslararası iş birlikleri ve kültürel alışveriş için yeni kapılar açar. Tiyatrolar, farklı ülkelerdeki sanatçılarla iş birliği yapabilir ve dünya çapında sanatsal projeler geliştirebilirler.

Bu raporun amacı bir online mobil uygulamayı, özel bir tiyatro için yapmaktır.

## Gelişme

Bu projede Flutter kullanarak tiyatro koltuğu seçme uygulaması geliştirildi. Uygulama, kullanıcıların mevcut tiyatroları görüntülemesine, koltuk seçmesine ve bilet rezervasyonu yapmasına olanak tanır.

* Bilet ekranı :

Tiyatro etkinliklerini duyurmak amacıyla yazılan bu uygulamada, yönetici (admin) ve kullanıcı (user) olmak üzere iki tip kullanıcı yetkisi oluşturulmuştur. İki kullanıcının da bilet alma yetkisi vardır.

Tiyatrolar sayfasında seçilen bir tiyatronun üzerine tıklanınca bilet ekranı görünür. Bilet ekranında tiyatronun adı, yeri, kayıtlı resmi, yapımcısı gibi bilgiler tiyatro ekranından çekilir. Ekranda bilet kodu oluşturulur.

‘Koltuk Seç’ butonu ile koltuk ekranına gidilirken ‘Bilet Al’ ekranı ile bütün değişkenler tamamlanır ve biletin son hali görünür.

Koltuk ekranına geçilmeden önce bir pop-up ekranında etkinliğin hangi salondan alınacağı seçilir.

* Koltuk ekranı :

Bilet ekranında pop-up ile seçilmiş olan salon numarası doğru salondan bilet alınması için miras alınır. Koltuk seçme ekranında tiyatro salonunun dijital versiyonu görünür. Eğer daha önceden alınmış koltuklar var ise bunlar, salonun veri tabanından kontrol edilir. Dolu koltukların rengi ekranda kırmızı, boş koltukların rengi yeşil renkli görünür. Koltukların numaraları gridlerin üzerinde tanımlıdır. Bu gridlerin üzerine tıklandığında yer ayırılacak koltuklar seçilmiş olur. Tekrar basıldığında koltuklar seçimden kaldırılabilir. Hangi koltuğun seçili olduğu ekranın alt kısmında görünür.

‘Yer Ayırt’ butonuna basıldığında ise seçilen koltuk numaraları salonun veri tabanında dolu görünür. Bu sayede o koltuğa o anda başka bir kullanıcı bilet ayıramaz. Koltuk zaten dolu uyarısı gelir.

* Bileti Al ekranı:

Bilet Al ekranında seçilen etkinliğin adı, tarihi, salon numarası, koltuk numaraları ve bilet kodu görünür. Bu noktada bilet artık oluşmuştur. Veri tabanındaki ‘tickets’ tablosunda da bulunur.

Proje, her biri belirli bir amaca hizmet eden birkaç dart dosyasına ayrılmıştır. Ticket\_page.dart doyası bilet rezervasyon sayfasını yönetir. Theaters.dart dosyası mevcut tiyatroları listeler. Ticket\_style.dart dosyası biletlerin stilini tanımlar. Tickets.dart dosyası biletler için veri yapısını tanımlar. Seat.dart dosyası koltukların temsili ve işlevselliğini yönetir. App\_color.dart dosyası uygulamadaki renk temalarını tanımlar.

Ticket\_style.dart dosyası, uygulamada bilet bilgilerinin görüntülenmesi için kullanılan metin stillerinin tanımlanmasından sorumludur. Bu dosyada, biletlerde olması planlanan değişkenler theater.dart dosyasından aktarılır. Bu sayede ilgili verilere erişilir. Sanal bir bilet görüntüsü vermek amacıyla Flutter'ın TextStyle sınıfını kullanır. Çeşitli çizgi fonksiyonları ile bilet şekli çizilir. Sayfada biletin öncelikli eşsiz değişkeni olacak olan bilet kodu oluşturulur.

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

*Şekil 1-2 ticket\_style.dart dosyası*

Tickets.dart dosyası, uygulamadaki bilet konseptini modelleyen bir Ticket sınıfını tanımlar. ‘tickets’ tablosuna veri eklenebilmesi için oluşturulmuştur. Tickets tablosu ‘ticket\_id’, ’ticket\_type’, ’price’, ’event\_id’, ’ticket\_date’, ‘tevent\_name’, ‘tvenue’, ‘tevent\_image’, ‘user\_email’, ‘ticket\_no’, ‘reserved\_seats’, ‘salon’ sütunlarından oluşuyor. ‘ticket\_id’ primary key, ‘event\_id’ ise events tablosu ile bağlantılı, id değerleri events’tan gelir. ‘saloonone’ ve ‘saloontwo’ tablolarında birbirine bağlı üç primary key bulunur. ‘event\_name’, ‘seat\_no’, ‘event\_date’ sütunları birbirine bağlı. Bunun böyle tanımlanmasının sebebi aynı koltuğa iki bilet kesilmesini önlemek. Saloonone ve saloontwo tablolarında koltuğun dolu oldup olmadığını tutan bir ‘seat\_state’ sütunu bulunuyor.

Seat.dart dosyasında ekranda görünen salon koltukları tasarlanır. Şekillerin düzeni, rengi, font çeşitleri belirlenir. Koltuklar önce uygulamanın arayüzünde oluşturulur. Sonradan veritabanı işlemleri devreye girer. ‘saloonone’ ve ‘saloontwo’ tablolarında daha önceden ayırtılmış koltuk bilgileri tutulur. Bu tablolardan hangi verilerin çekilmesi gerektiği; bilet sayfasından çekilen etkinliğin adı, tarihi ve salonu ile belli olur. Bu tarihler ile çakışan satırlar var ise o satırlardaki koltuk numaraları ekranda kırmızıya döner. Ticket\_style.dart dosyasında tanımlanmış bilet kodu sayfa açılır açılmaz oluşturulur.

A screen shot of a computer program

Description automatically generated

*Şekil 3 ticket.dart dosyası*

Ticket\_style dosyasında bulunan değişkenler Seat widget’ına aktarılır. Bu biletin ayırılabilmesi için gereklidir. Yer ayrılmak istendiğinde doğru etkinlik bilgileri ile yer ayrılması, TicketStyle widget’ından alınan bilgiler sayesinde mümkün olur.

A screenshot of a computer program

Description automatically generated

*Şekil 4 Koltuk Seç ve Bilet al butonları*

*A screenshot of a computer

Description automatically generated*A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated*Şekil 5-6 seat.dart dosyası*

*Şekil 7 tickets tablosu*

*A screenshot of a computer

Description automatically generatedŞekil 8 saloonone tablosu*

*A screenshot of a computer

Description automatically generatedŞekil 9 saloontwo tablosu*

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated A screenshot of a game

Description automatically generated

A screenshot of a phone

Description automatically generated*Şekil 10-11-12 Bilet ekranı, Salon seçim ekranı, Koltuk seçim ekranı*

*Şekil 13 Bilet oluşturma ekranı*

Giriş ekranı, kullanıcılardan e-posta adresi ve şifre talep eder. Girilen bilgiler Supabase'in kimlik doğrulama API'sine gönderilir ve başarılı bir giriş durumunda oturum tokeni döndürülür. Bu token, uygulamada kullanıcının oturum açık kaldığını gösterir. Ayrıca, uygulamada kayıtlı kullanıcıların bilgileri Supabase'in 'users' tablosunda saklanır. Tiyatrolar ekranında, Supabase veritabanından çekilen etkinlikler listelenir. Etkinlikler 'EventWidget' sınıfı aracılığıyla gösterilir ve her etkinlik için etkinlik adı, türü, yapımcısı, tarihi ve resim içeren bir kart oluşturulur. Yeni etkinlik ekleyebilme yetkisi yalnızca yönetici kullanıcılara tanınır. Etkinliklerin görüntülenmesi ve yönetilmesi Supabase'in veri yönetimi özelliklerini kullanarak gerçekleştirilir.

Uygulama sadece tarihi geçmemiş etkinlikleri listeler. Bu, SQL sorgularına eklenen bir koşulla sağlanır, böylece geçmiş tarihli etkinlikler kullanıcı arayüzünde görünmez. Bu özellik, kullanıcılara güncel ve ilgili bilgilerin sunulmasını sağlar.

Tiyatro etkinliklerini duyurmak amacıyla yazılan bu uygulamada, yönetici (admin) ve kullanıcı (user) olmak üzere iki tip kullanıcı yetkisi oluşturulmuştur. İki kullanıcının da bilet alma yetkisi vardır. Tiyatrolar sayfasında seçilen bir tiyatronun üzerine tıklanınca bilet ekranı görünür. Bilet ekranında, tiyatronun adı, yeri, kayıtlı resmi, yapımcısı gibi bilgiler tiyatro ekranından çekilir. Ekranda bilet kodu oluşturulur.

Bilet ekranında pop-up ile seçilmiş olan salon numarası doğru salondan bilet alınması için miras alınır. Koltuk seçme ekranında tiyatro salonunun dijital versiyonu görünür. Eğer daha önceden alınmış koltuklar var ise bunlar salonun veri tabanından kontrol edilir. Dolu koltukların rengi ekranda kırmızı, boş koltukların rengi yeşil renkli görünür. Koltukların numaraları gridlerin üzerinde tanımlıdır. Bu gridlerin üzerine tıklandığında yer ayırılacak koltuklar seçilmiş olur. Tekrar basıldığında koltuklar seçimden kaldırılabilir. Hangi koltuğun seçili olduğu ekranın alt kısmında görünür.

App\_colors.dart dosyası uygulamada bilet bilgilerinin görüntülenmesi için kullanılan metin stillerinin tanımlanmasından sorumludur. Bu dosyada, biletlerde olması planlanan değişkenler theater.dart dosyasından aktarılır. Bu sayede ilgili verilere erişilir. Sanal bir bilet görüntüsü vermek amacıyla Flutter'ın TextStyle sınıfını kullanır.

Flutter, Google tarafından geliştirilen açık kaynaklı bir UI yazılım geliştirme kitidir. Mobil, web ve masaüstü uygulamaları geliştirmek için kullanılır. Flutter'ın benimsenmesinin ardındaki en büyük nedenlerden biri, aynı kod tabanını kullanarak çeşitli platformlarda uygulamalar geliştirme yeteneğidir. Bu, geliştirme süresini ve maliyetini önemli ölçüde azaltır.

Hızlı Geliştirme: Flutter'ın "hot reload" özelliği, kod değişikliklerinin anında görülebilmesini sağlar. Bu, geliştirme sürecini hızlandırır ve geliştiricilerin hızlıca iterasyon yapmasına olanak tanır.

Yerel Performans:Flutter, doğrudan makine diline derlenir ve bu da yüksek performanslı uygulamalar oluşturulmasına olanak tanır.

Zengin Widget Kütüphanesi: Flutter, önceden oluşturulmuş geniş bir widget kütüphanesi sunar. Bu widget'lar, platformdan bağımsız olarak çalışır ve her iki platformda da (iOS ve Android) yerel bir görünüm ve his sunar.

UI Tasarımı: Flutter, uygulamanızın kullanıcı arayüzünü (UI) oluşturmak için bir dizi özelleştirilebilir widget sunar. Bu widget'lar, yerleşim, stil ve animasyonları kolayca yönetmenizi sağlar.

State Management: Flutter'da durumu yönetmek için çeşitli yaklaşımlar bulunur. En yaygın kullanılan yöntemler arasında Provider, Riverpod ve Bloc yer alır.

Supabase, mevcut uygulamalarınıza gerçek zamanlı veritabanı, kimlik doğrulama ve depolama eklemek için kullanılan açık kaynaklı bir arka uç platformudur. Firebase'e açık kaynaklı bir alternatiftir ve Postgres veritabanı üzerine inşa edilmiştir.

Gerçek Zamanlı Veritabanı: Supabase, veritabanınızda yapılan değişiklikleri gerçek zamanlı olarak uygulamalarınıza iletebilir.

Kimlik Doğrulama: Kullanıcı kayıtları, giriş işlemleri ve üçüncü taraf kimlik doğrulama sağlayıcıları (Google, Facebook, GitHub vb.) ile entegre çalışır.

Depolama: Dosya depolama ve yönetim hizmeti sunar, böylece kullanıcı dosyalarını güvenli bir şekilde saklayabilirsiniz.

Veritabanı Yönetimi: Supabase, PostgreSQL'i temel alır ve bu da güçlü ve ölçeklenebilir bir veritabanı çözümü sunar. SQL sorguları ile veri işleme, saklama ve yönetim işlemlerini kolayca gerçekleştirebilirsiniz.

Kimlik Doğrulama: Supabase Auth, kullanıcıların güvenli bir şekilde kimlik doğrulamasını sağlar ve oturum yönetimini kolaylaştırır.

Depolama: Supabase Storage, dosyaları güvenli bir şekilde yüklemenizi ve yönetmenizi sağlar. Uygulamanızdan medya dosyalarına kolayca erişebilir ve bunları kullanıcılarınıza sunabilirsiniz.

Dart, Google tarafından geliştirilen açık kaynaklı bir programlama dilidir. Flutter uygulamaları geliştirmek için kullanılan birincil dildir. Dart, hem istemci tarafında (UI) hem de sunucu tarafında (arka uç) çalışabilir.

Basit ve Hızlı: Dart, öğrenmesi kolay bir sözdizimine sahiptir ve hızlı derleme süreleri sunar.

Objektif Yönelimli: Dart, nesne yönelimli programlama özelliklerini destekler, bu da büyük ve karmaşık uygulamaların yönetimini kolaylaştırır.

Asenkron Programlama: Dart, asenkron işlemleri yönetmek için güçlü bir desteğe sahiptir. Bu, uygulamanızın performansını ve kullanıcı deneyimini artırır.

Flutter ile Entegrasyon: Dart, Flutter ile sorunsuz bir şekilde çalışır ve Flutter SDK'nın tüm gücünden yararlanmanıza olanak tanır.

Web ve Mobil Geliştirme: Dart, hem web hem de mobil uygulamalar geliştirmek için kullanılabilir, bu da geliştiricilere platformlar arası uygulamalar oluşturma esnekliği sağlar.

Projenin kullanıcı arayüzü tamamen Flutter kullanılarak geliştirildi. Flutter'ın sağladığı hızlı geliştirme süreci ve yerel performans avantajları, projeyi kullanıcı dostu ve performanslı hale getirdi. Dart dili ise, Flutter ile uyumlu çalışarak projenin temel programlama dili olarak kullanıldı. Bu ikili, projenin temel taşlarını oluşturdu.

Projenin arka uç hizmetleri için Supabase kullanıldı. Gerçek zamanlı veritabanı ve kimlik doğrulama özellikleri, uygulamanın güvenli ve hızlı çalışmasını sağladı. Supabase'in sağladığı PostgreSQL tabanlı veritabanı, güçlü ve esnek veri yönetimi çözümleri sundu.

Kalıtım, nesne yönelimli programlamanın (OOP) temel taşlarından biridir. Kalıtım, bir sınıfın başka bir sınıfın özelliklerini ve metotlarını devralmasına olanak tanır. Bu, kod tekrarını azaltır ve daha esnek ve yeniden kullanılabilir yazılım bileşenleri oluşturmayı sağlar. Kalıtım, yazılım mühendisliğinde daha organize ve modüler bir kod yapısı elde etmek için kullanılır.

Bu projede, çeşitli sınıflar arasında kalıtımı kullanarak özellikleri ve işlevsellikleri paylaşmak için kalıtım teknolojisinden yararlanıldı.

`ticket\_style.dart` dosyasında tanımlanan `TicketStyle` sınıfı, biletlerin nasıl görüneceğini tanımlar. `tickets.dart` dosyasında tanımlanan `Ticket` sınıfı ise, bilet bilgilerinin modelini içerir ve `TicketStyle` sınıfının stil özelliklerini kullanır. `TicketStyle` sınıfı, biletlerin metin stillerini tanımlar. Bu sınıf, bilet numarası ve koltuk durumu gibi metinlerin nasıl görüneceğini belirler.

`Ticket` sınıfı, bilet modelini tanımlar ve `TicketStyle` sınıfının özelliklerini kullanarak biletin görünümünü belirler.

`Ticket` sınıfı, `Seat` sınıfının bir örneğini içerir ve biletin fiyatını belirler. `TicketStyle` sınıfı ise, biletin kullanıcı arayüzünde nasıl görüneceğini tanımlar. Bu iki sınıf arasındaki ilişki, biletlerin hem veri modelini hem de stilini belirler ve böylece daha organize bir yapı sağlar.

Projedeki sayfalar arasında veri aktarımı, kalıtım ve Flutter'ın widget yapısı kullanılarak gerçekleştirilir. `ticket\_page.dart`, `theaters.dart` ve `seat.dart` dosyaları, sayfalar arasında veri aktarımını ve kullanıcı etkileşimini sağlar.

`TicketPage` sınıfı, `Ticket` nesnesini parametre olarak alır ve bu bilgiyi ekranda görüntüler.

`TheatersPage` sınıfı, mevcut biletleri listeleyen bir sayfayı tanımlar. Kullanıcı bir bileti seçtiğinde, `TicketPage` sayfasına yönlendirilir ve seçilen biletin detayları gösterilir.

`TicketPage` sınıfı, `TheatersPage` sınıfından seçilen bilet verisini alır ve bu veriyi kullanıcıya gösterir. Bu, sayfalar arasında veri aktarımını ve kullanıcı etkileşimini sağlar.

Kalıtım, projenin daha modüler ve yönetilebilir olmasını sağlar. Kod tekrarını azaltır ve sınıflar arasında ortak işlevlerin paylaşılmasını kolaylaştırır. Ayrıca, sınıflar arasında açık bir hiyerarşi oluşturarak kodun anlaşılmasını ve bakımını kolaylaştırır.

Kod Tekrarını Azaltma: Kalıtım, aynı kodun birden fazla yerde yazılmasını önler.

Modülerlik: Sınıflar arasında ortak işlevler paylaşılabilir, bu da kodun modüler olmasını sağlar.

Bakım Kolaylığı: Kalıtım, kodun anlaşılmasını ve bakımını kolaylaştırır, çünkü kod yapısı daha düzenli ve hiyerarşik olur.

Ticket ve TicketStyle sınıflarında bilet bilgilerini ve stillerini yönetmek için kalıtım kullanıldı. Kullanıcı etkileşimini ve veri aktarımını sağlamak için kalıtım ve Flutter widget yapısı kullanıldı.

Bu detaylı açıklamalar, projede kullanılan kalıtım teknolojilerini ve bunların nasıl uygulandığını kapsamlı bir şekilde açıklar. Raporunuzda bu metinleri kullanarak projenizin teknolojik altyapısını vurgulayabilirsiniz.

A screenshot of a cell phone

Description automatically generatedA screenshot of a phone

Description automatically generated

## *Şekil 10-11 Giriş ekranı ve Tiyatrolar ekranı*

<https://github.com/Nur-betul/theater_app.git>

## Sonuç

Bu rapor, bir tiyatro salonunun etkinliklerini online mobil uygulama aracılığıyla duyurmasının önemini ve etkinliğini ele almaktadır. Projede geliştirilen mobil uygulama, tiyatroların dijital dönüşümüne katkı sağlamak ve geniş bir kitleye ulaşmasını kolaylaştırmak amacıyla tasarlanmıştır. Uygulama, kullanıcıların tiyatro etkinlikleri hakkında kolayca bilgi edinmelerine, bilet satın almalarına ve etkinlik programlarına erişmelerine olanak tanır.

Projede geliştirilen mobil uygulama, tiyatroların dijital dünyada varlık göstermesini sağlayabilir ve genç nesiller arasında tiyatroya olan ilgiyi artırabilir. Dijital biletler ve kolay erişim özellikleri ile hem kullanıcı deneyimi iyileştirilmiş hem de çevre dostu bir alternatif sunulmuştur.

Netice olarak, bu tarz bir uygulama, tiyatroların dijital dönüşümüne önemli bir katkı sağlayabilir ve sanatsal etkinliklerin daha geniş bir kitleye ulaşmasını mümkün kılar.

Tiyatro salonlarının, teknolojiyi benimseyerek sanatsal ifadelerini ve etkileşimlerini genişletmeleri, dijital çağda sanatın sürdürülebilirliği ve geniş kitlelere ulaşımı açısından hayati önem taşımaktadır.

Sonuç olarak, bu proje, mobil teknolojinin tiyatroların erişilebilirliğini ve etkileşimini nasıl artırabileceğinin somut bir örneğini sunmaktadır. Projede geliştirilen uygulama, hem tiyatrolar için hem de sanatseverler için değerli bir kaynak oluşturmuştur. Bu başarı, dijitalleşmenin tiyatrolar için sadece bir seçenek değil, aynı zamanda gelecekteki başarının anahtarı olduğunu kanıtlamaktadır.

# KAYNAKÇA

# <https://supabase.com/docs/guides/getting-started/quickstarts/flutter>

# <https://supabase.com/docs/guides/getting-started/tutorials/with-flutter>

# <https://supabase.com/docs/reference/dart/insert>

# <https://supabase.com/docs>