T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ

2024-2025 EĞİTİM ÖĞRETİM YILI

GÜZ DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİTABANI YÖNETİM SİSTEMLERİ DERSİ

PROJE-FİNAL DÖKÜMANI

|  |
| --- |
| Tüm sorular soru metnini altındaki alanlara cevaplandırılacaktır. Bu belgenin formatını bozmayınız, sadece gerekli alanları doldurunuz. |

1. Gerçekleştirdiğiniz veri tabanı projesi için grup arkadaşlarınızın isimlerini yazınız ve projenize ait veri tabanı/diğer yazılım bileşenleri hakkında bilgi veriniz. (5 p)

*Enes Çakmak, 171422012*

*Emre Şık, 171422002*

*Engin Çetintaş, 170422026*

Bu proje kapsamında, MySQL kullanılarak ilişkisel bir veri tabanı tasarlanmış ve 3NF (Üçüncü Normal Form) prensiplerine göre optimize edilmiştir. Veri tabanı; kullanıcılar, sanatçılar, şarkılar, albümler, türler, çalma listeleri ve duygular gibi ana tablolar ile bu tablolar arasındaki ilişkileri yöneten bağlantı tablolarından oluşmaktadır.

Web arayüzü tasarımı için HTML, CSS, TailwindCSS ve JavaScript kullanılmıştır. HTML, sayfa yapısını oluştururken; CSS ve TailwindCSS, görsel düzen ve stil kazandırmış; JavaScript ise dinamik ve interaktif özellikler (şarkı ekleme, çalma listesi yönetimi) eklemek için tercih edilmiştir. Arayüz, kullanıcıların kolayca müzikleri görebileceği ve çalma listeleri oluşturabileceği kullanıcı dostu bir deneyim sunmaktadır.

1. Gerçekleştirdiğiniz veri tabanı projesi için proje dokümanınızı ve dosyalarınızı içeren herkese açık github bağlantılarınızı paylaşınız. (5 p)

<https://github.com/enesscakmak/music-web-app>

1. Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (10 p)

Bu projenin amacı, bir müzik akış uygulaması için veri tabanı tasarlamaktır. Kullanıcılar, şarkılar, albümler, çalma listeleri, türler, duygular, beğeniler ve takipçi ilişkileri gibi verilerin yönetilebildiği bir sistem sunmak hedeflenmiştir. Tasarım, veri tekrarını önlemek, verilerin esnek bir şekilde yönetilmesini sağlamak ve performansı artırmak için geliştirilmiştir.

Bu sistem, kullanıcıların:

* Şarkı, albüm ve sanatçıları listelemesini
* Kendi çalma listelerini oluşturmasını ve yönetmesini
* Yeni şarkılar, albümler ve sanatçılar eklemesini
* Kayıt olmasını ve giriş yapabilmesini sağlar

ekran görüntüsü, metin, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Tasarladığınız veri tabanı mimarisinde hangi tablo ve ilişkileri kullanıldığınızı açıklayınız. (10 p)

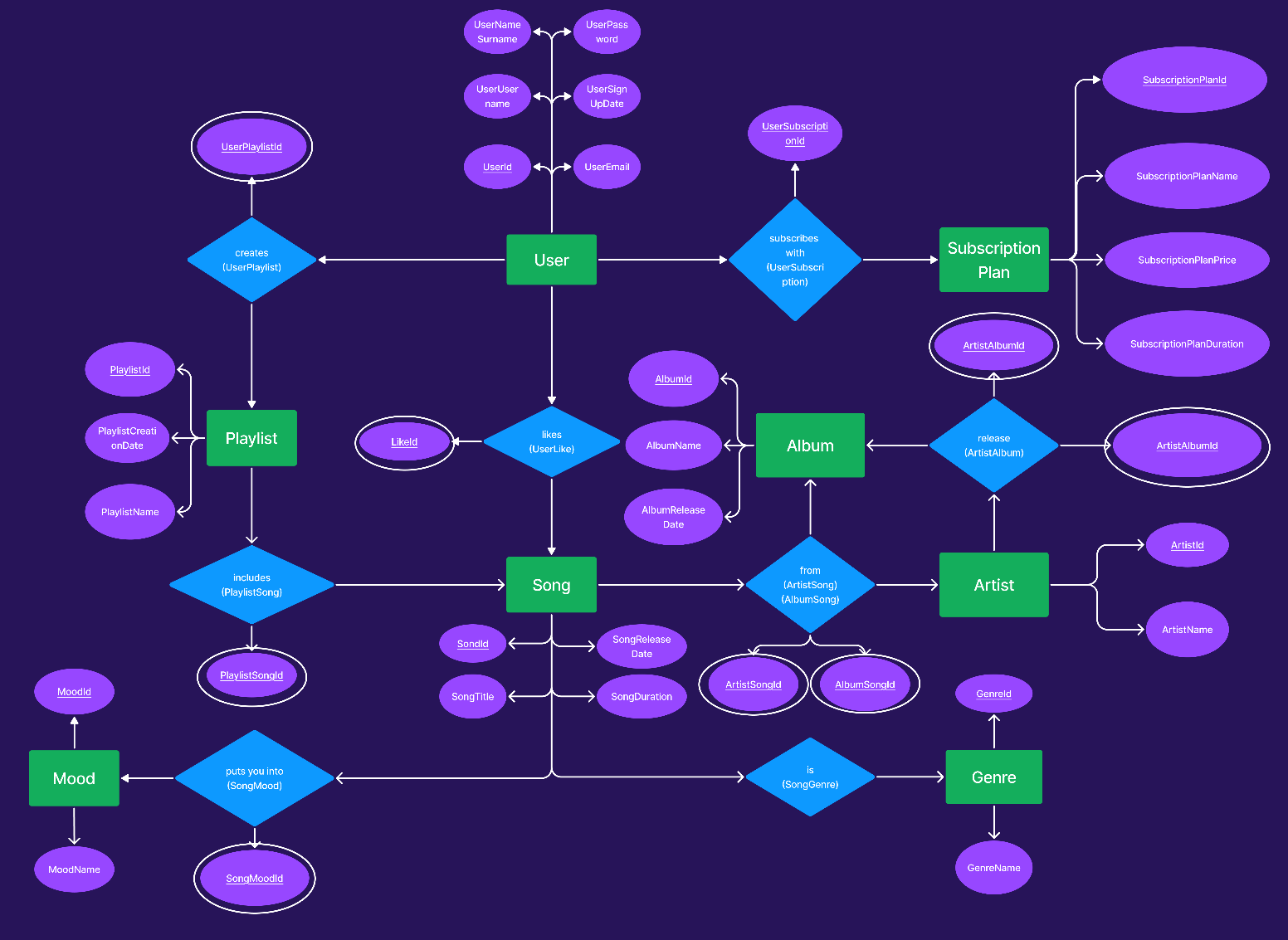
Veri Tabanı Mimarisi  
Proje, ilişkisel bir veri tabanı mimarisi kullanılarak tasarlanmıştır. Sistem şu tabloları içerir:

Ana Tablolar

1. Users (Kullanıcılar): Kullanıcı bilgilerini ve kimlik doğrulama detaylarını tutar.
2. Artists (Sanatçılar): Sanatçılara dair bilgileri saklar.
3. Songs (Şarkılar): Şarkılara ait temel bilgileri içerir.
4. Albums (Albümler): Albüm bilgilerini içerir ve şarkılarla ilişkilidir.
5. Genres (Türler): Şarkı türlerini tanımlar.
6. Playlists (Çalma Listeleri): Kullanıcıların oluşturduğu çalma listelerini yönetir.
7. Moods (Duygular): Şarkıların duygu durumlarına göre sınıflandırılmasını sağlar.
8. SubscriptionPlans (Abonelik Planları): Kullanıcı abonelik seçeneklerini tutar.

İlişki Tabloları

1. UserLike (Kullanıcı-Begeni): Kullanıcıların beğendiği şarkıları ilişkilendirir.
2. SongMood (Şarkı-Duygu): Şarkıların duygularla bağlantısını kurar.
3. SongGenre (Şarkı-Tür): Şarkıların tür bilgilerini ilişkilendirir.
4. PlaylistSong (Çalma Listesi-Şarkı): Çalma listeleri ile şarkılar arasındaki bağlantıyı yönetir.
5. UserPlaylist (Kullanıcı-Çalma Listesi): Kullanıcılar ve çalma listeleri arasındaki ilişkiyi yönetir.
6. UserSubscription (Kullanıcı-Abonelik): Kullanıcıların aktif ve geçmiş aboneliklerini tutar.
7. AlbumSong (Albüm-Şarkı): Albümler ve şarkılar arasındaki ilişkiyi tutar.
8. ArtistSong (Sanatçı-Şarkı): Sanatçılar ve şarkılar arasındaki ilişkiyi tutar.
9. ArtistAlbum (Sanatçı-Albüm): Sanatçılar ve albümler arasındaki ilişkiyi yönetir.
10. Veri tabanı ER (Entity Relationship) diagramının bilgisayar ortamında çizilmiş halini paylaşınız. (Ara raporda eksik kısımlar bu raporda giderilmelidir ve ER çizme programlarından faydalanıbilir. Elle çizim, çizip fotoğrafını çekme vb. kabul edilmeyecektir.) (10 p)

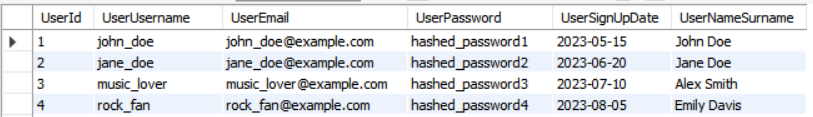


1. Herhangi iki tablonuz için DDL (create) kodları yazılmalıdır. (10 p)

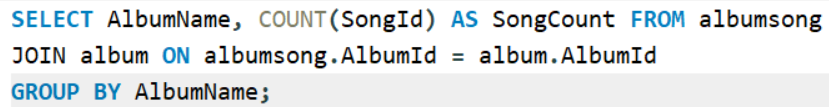
CREATE TABLE User (  
 UserId INT AUTO\_INCREMENT PRIMARY KEY,  
 UserUsername VARCHAR(100),  
 UserEmail VARCHAR(150),  
 UserPassword VARCHAR(255),  
 UserSignUpDate DATE,  
 UserNameSurname VARCHAR(100)  
);

CREATE TABLE Song (  
 SongId INT AUTO\_INCREMENT PRIMARY KEY,  
 SongTitle VARCHAR(150),  
 SongDuration TIME,  
 SongReleaseDate DATE,  
 AlbumId INT,  
 ArtistId INT,  
 FOREIGN KEY (AlbumId) REFERENCES Album(AlbumId),  
 FOREIGN KEY (ArtistId) REFERENCES Artist(ArtistId)  
);

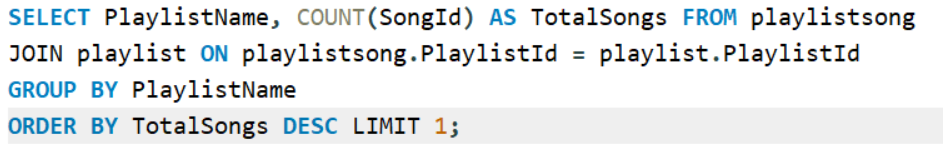
1. 5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)
2. INSERT INTO User (UserUsername, UserEmail, UserPassword, UserSignUpDate, UserNameSurname) VALUES ('jane\_doe', 'jane.doe@example.com', 'securepassword123', '2024-12-20', 'Jane Doe');
3. INSERT INTO Playlist (PlaylistName, PlaylistCreationDate, UserId) VALUES ('Jane\'s Chill Vibes', '2024-12-21', LAST\_INSERT\_ID());
4. INSERT INTO SongGenre (SongId, GenreId) VALUES (1, 2), (1, 3);
5. INSERT INTO Album (AlbumName, AlbumReleaseYear, ArtistId) VALUES ('Greatest Hits 2024', 2024, 1);
6. INSERT INTO UserSubscription (UserId, SubscriptionPlanId) VALUES (3, 2);



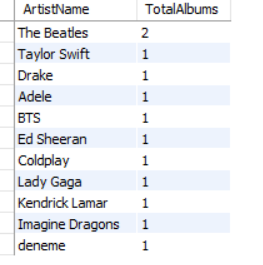
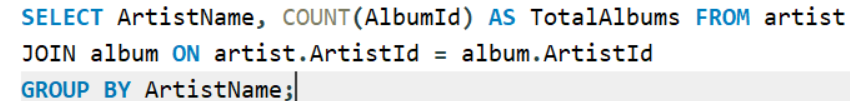
1. Projenize ait kendi belirlediğiniz 10 adet SQL sorgusu yazınız, sorguların amacını ve sonuç çıktısını da lütfen ekleyiniz. (Açıklama: Sorgular Select deyimleri ve gruplama fonksiyonlarını HAVING deyimini (min, max, avg, count gibi) ve join deyimlerini (en az iki tablo ile birleştirme sorgusu) içerecek şekilde basitten karmaşığa doğru gitmelidir. Proje sunum anında veri tabanınıza ait sorular SQL ortamında gösterilecek ve açıklanacaktır. Raporunuzda ise sorgular, sorguların cevap ve sonuçlarının ekran görüntüsü olarak paylaşılması beklenmektedir. (10 p)
2. Albümlerdeki şarkı sayısı:



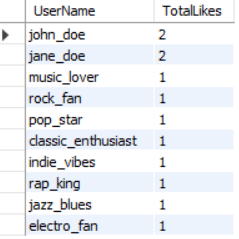
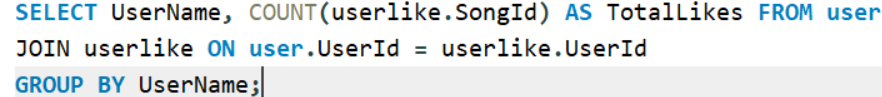
1. En çok şarkı eklenen çalma listesi:



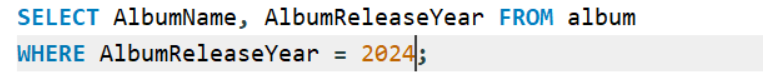
1. Her sanatçının toplam albüm sayısı:

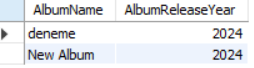


1. Kullanıcıların şarkı beğenme sayısı:

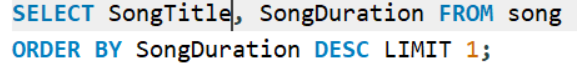


1. Belirli bir yılda (örneğin 2024) çıkan albümler

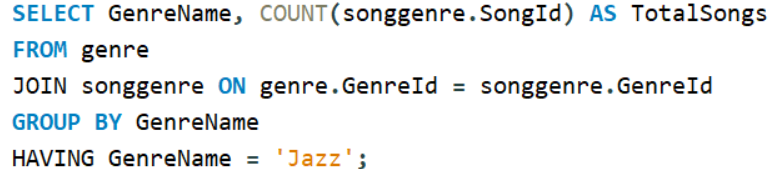




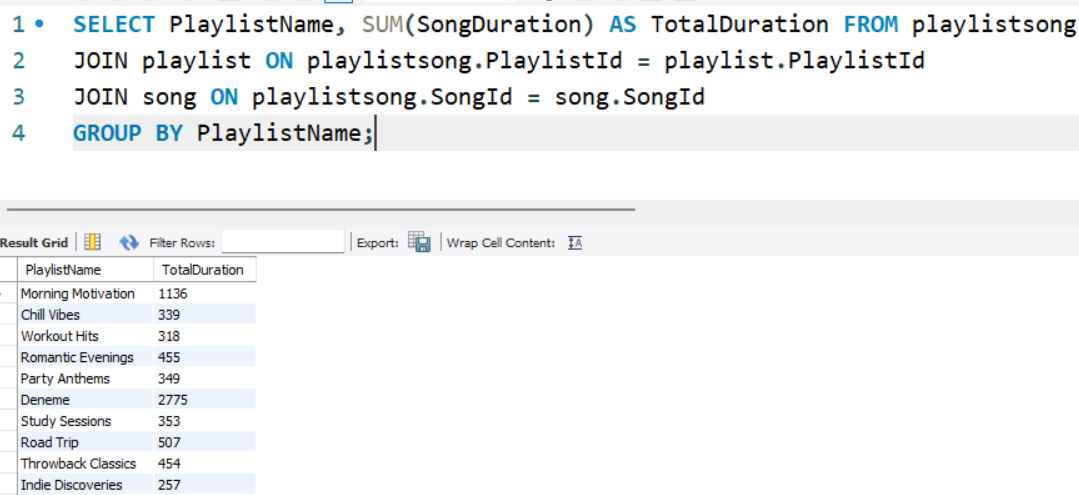
1. En uzun şarkının bilgisi



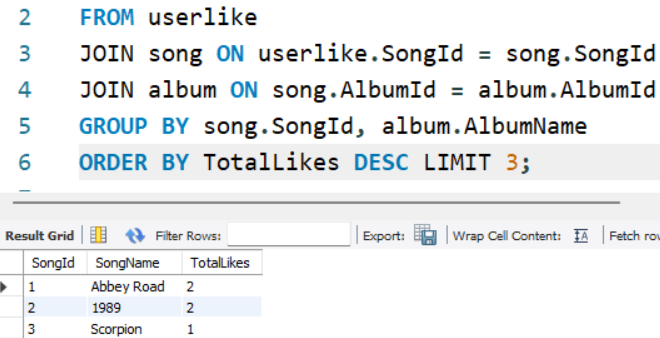
1. Belirli bir türün şarkı sayısı:



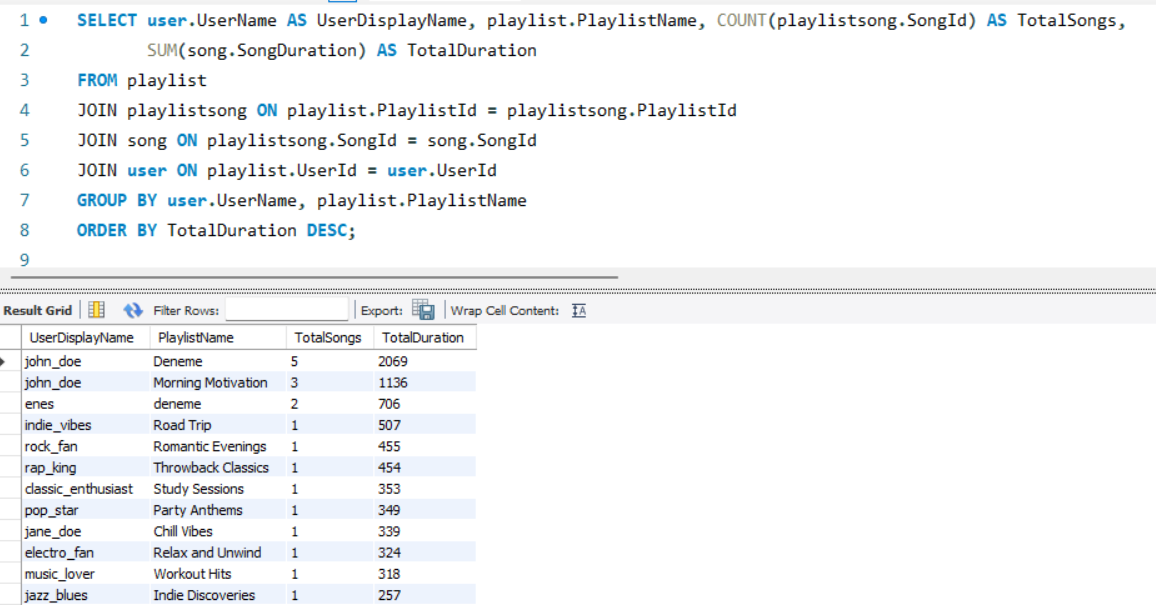
1. Her çalma listesinde toplam çalma süreleri:



1. En çok beğenilen şarkılar:

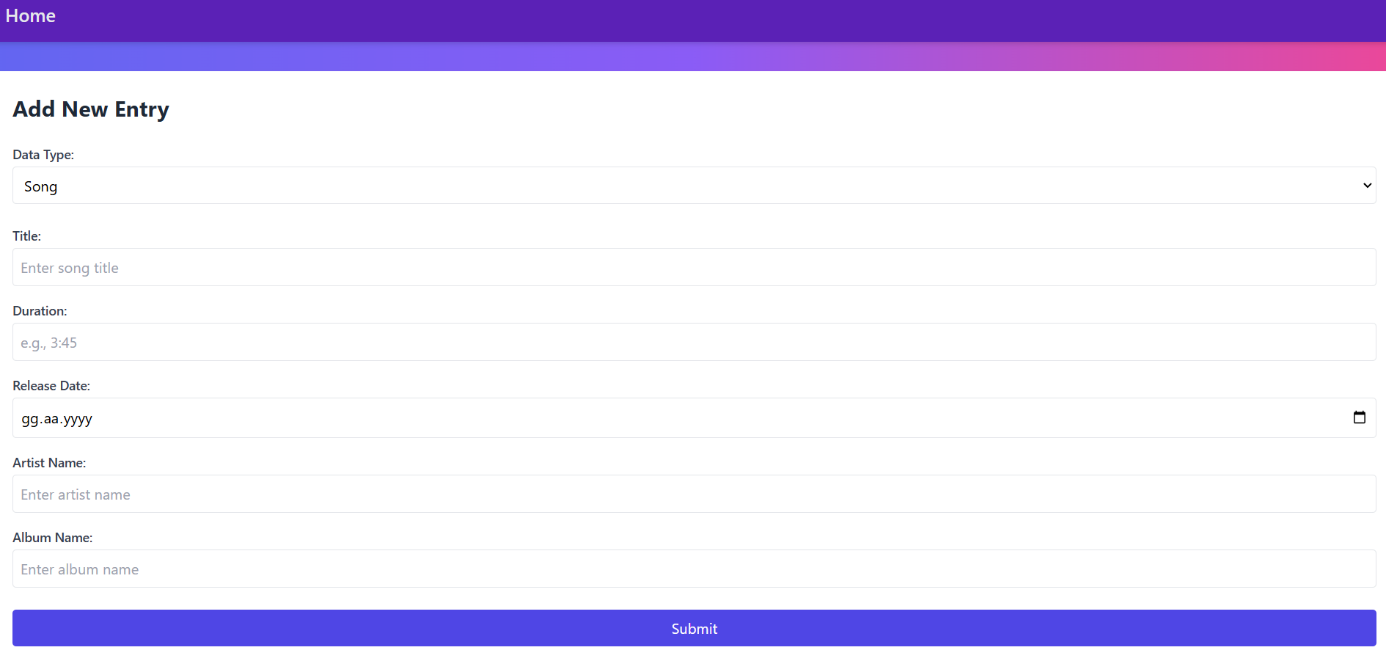


1. Kullanıcı bazında çalma listesi ve şarkı analizleri:



1. Eğer gerçekleştirmiş iseniz, veri tabanı bağlama ve uygulama geliştirme aşamalarınızı kısaca açıklayarak, kullanıcı ara yüz ekranından bir örnek veriniz. Ve geliştirdiğiniz ara yüzü anlatınız. (10 p)

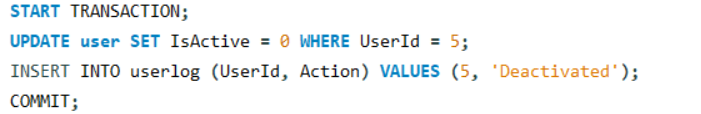
Uygulamamız Flask kütüphanesi ile Html Css JavaScript kullanılarak geliştirildi. Kullanıcılar şarkıları görebilir, çalma listeleri oluşturabilir ve şarkıları çalma listelerine ekleyebilir. Aşağıda bir arayüz ekran görüntüsü verilmiştir:



1. Eğer veri tabanı bağlama işlemini gerçekleştirmemiş iseniz VTYS sistemlerinde Transaction nedir açıklayınız ve çalışmanızdan bir Transaction örneği veriniz. (10 p)

***Örnek olması amacıyla mevcut olmayan tablolar ve sütunlar kullanılmıştır.***

Transaction, bir veritabanında bir grup işlemın bir bütün olarak yürütülmesi için kullanılır. Transaction doğruluğu sağlamak için ACID (Atomicity, Consistency, Isolation, Durability) özelliklerini destekler.



Buradaki Transaction ile:

Bir Kullanıcıyı Devre Dışı Bırakma:

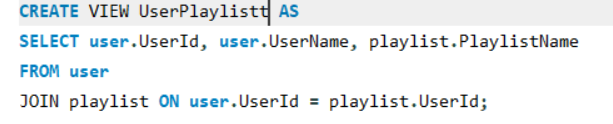
* user tablosunda belirli bir kullanıcıyı devre dışı bırakmak için IsActive sütununu 0 yapar.

Kullanıcının İşlemini Kaydetme:

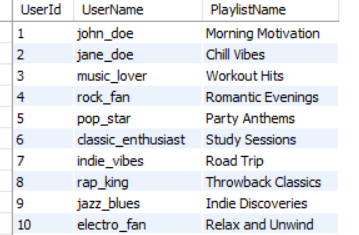
* userlog tablosuna, devre dışı bırakılan kullanıcının kimliği ve yapılan işlemle ilgili bir kayıt ekler.

1. View nedir açıklayınız ve bir adet view, bir adet saklı yordam (Stored Procedute) ifadesine ait SQL deyimlerinin sorgusunu ve cevabını yazınız.  
   (10 p)

View, veritabanında bir veya birden fazla tablodan türetilmiş sanal bir tablo veya sorgudur. Fiziksel olarak verileri saklamaz, ancak altındaki tablolardaki verilere sorgu aracılığıyla erişim sağlar.







Oluşturduğumuz View, kullanıcıların kimlik bilgilerini ve çalma listesi adlarını ilişkilendiren bir görünüm sağlar.

****Stored Procedure (Saklı Yordam)****, veri tabanında saklanan ve ihtiyaç duyulduğunda çağrılabilen bir SQL komutları grubudur. Programlanabilir bir mantık sunar ve parametreler alarak dinamik işlemler gerçekleştirebilir. Genelde sık kullanılan SQL işlemlerini kolaylaştırmak ve daha düzenli bir kod yapısı sağlamak için kullanılır.

