

26 ARALIK 2021

WEB PROGRAMLAMA E-MUSIC PROJE

AD SOYAD:İSMAİL ÜÇEL

[HTTPS://GITHUB.COM/İSMAİLUCEL/WEBPROGRAMLAMA_PROJE](https://github.com/ismailucel/WebProgramlama_Proje)

NUMARA: G201210352

SINIF GRUBU: 2.ÖĞRETİM A GRUBU

İçindekiler

Proje Tanıtım.....	3
Veritabanı Modeli.....	4
Ekran Görüntüleri.....	5

NOT: PROJE BOYUTU 25 MB'DAN FAZLA OLDUĞU İÇİN SABİS'E YÜKLEYEMEDİM. AŞAĞIDAKİ GİTHUB LİNKİNDE PROJENİN AŞAMALARI VE PROJE FİNAL KLASÖRÜNDE DE PROJENİN TAMAMLANMIŞ HALİNİ BULABİLİRSİNİZ.

PROJE GİTHUB LİNKİ:

https://github.com/ismailucel/WebProgramlama_Proje

Proje Tanıtım:

Bu proje internetten şarkı satın almamıza olanak sağlayan (global benzeri olarak itunes gibi) bir e-commerce web sitesidir. Projede iki kullanıcı rolü vardır. Bu roller Admin ve Customer rolleridir. Admin'lerin admin paneli sayesinde şarkılara, kategorilere, sanatçılara ve bekleyen siparişlere erişimi vardır. Bu paneller aracılığıyla şarkılar, kategoriler, sanatçılar eklenebilir, düzenlenebilir, aktiflik ve deaktiflik durumları düzenlenebilir. Customer'ların verdiği siparişlerin detayları gözlenebilir ve bu siparişler kargoya verilebilir. Customer'lar şarkıları sepetlerine ekleyip sipariş verebilir. Stripe API'si sayesinde Customer'lar ödemelerini yapabilirler. Şarkı detaylarına bakabilir, stok durumlarını inceleyebilirler. Customer'lar sepetlerini inceleyebilir, alışverişe devam edebilir ya da satın alma işlemini gerçekleştirebilir.

Şarkıları görüntüleyebileceğimiz, düzenleyebileceğimiz ve silebileceğimiz web api ile yazılmış bir API uygulamamız da vardır. Postman aracılığıyla bu uygulamanın ekran görüntülerini ekleyeceğim.

Rol işlemleri Areas içinde yapılmıştır. Customer ve Admin klasörlerinin kendi özel controllerları ve view'leri vardır. Model klasörü içinde uygulama içerisinde kullandığımız yapılar vardır.

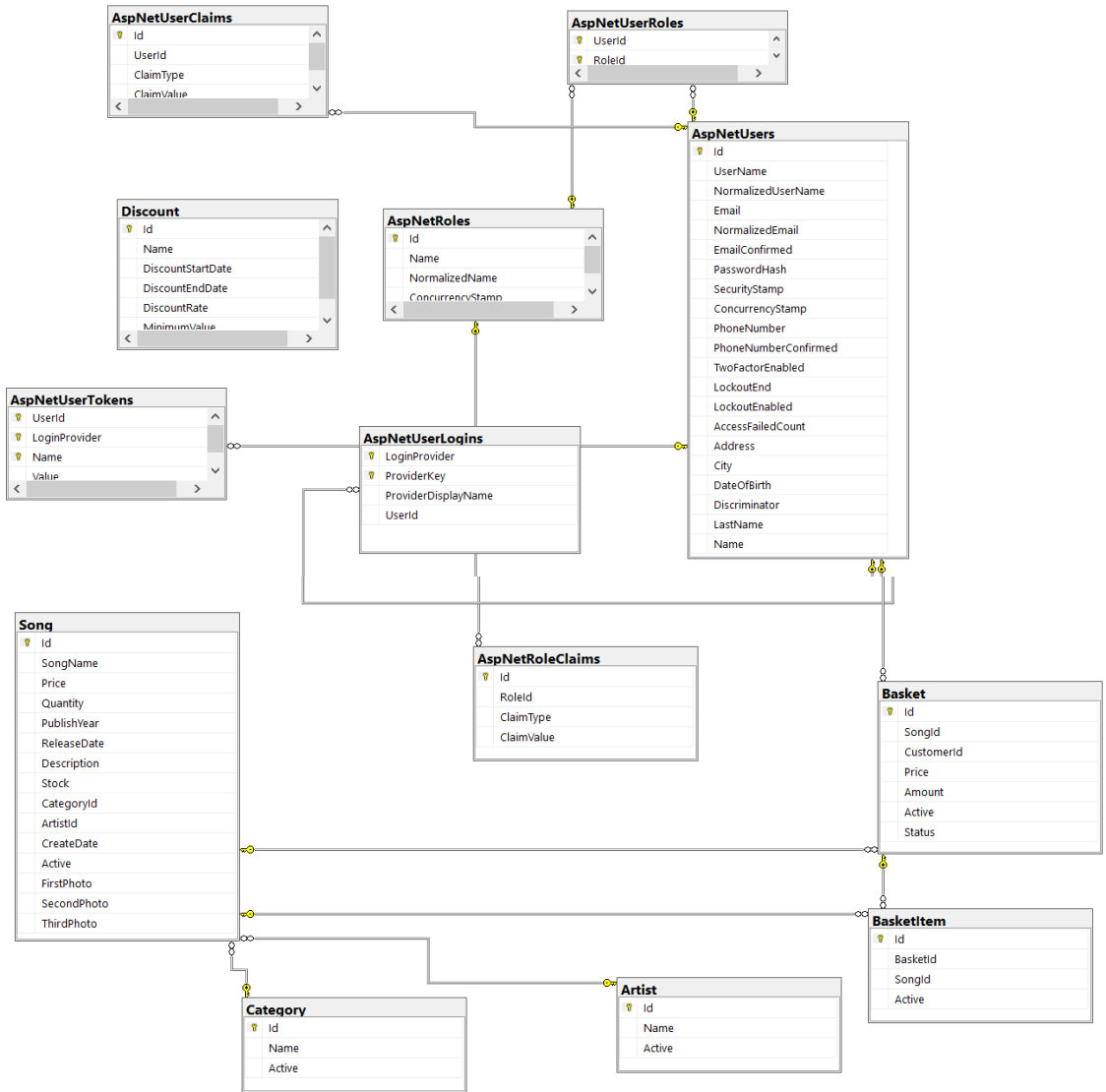
Controller klasörü içerisinde uygulama ilk açıldığında gelen sayfayı ve action'ları belirten HomeController'ı içerir.

Data klasörü içerisinde migration'lar, veritabanı başlatıcı sınıflar, stripe ayarları için oluşturulan sınıflar ve applicationdbcontext dosyası bulunur

Resources klasörü içerisinde çoklu dil için oluşturduğumuz resx dosyaları bulunur. Çoklu dil olarak İngilizce ve Türkçe vardır.

Diğer bir solution olan ApiG201210352 içinde ise yazdığım web api'nin dosyaları bulunmaktadır.

Veritabanı Modeli:



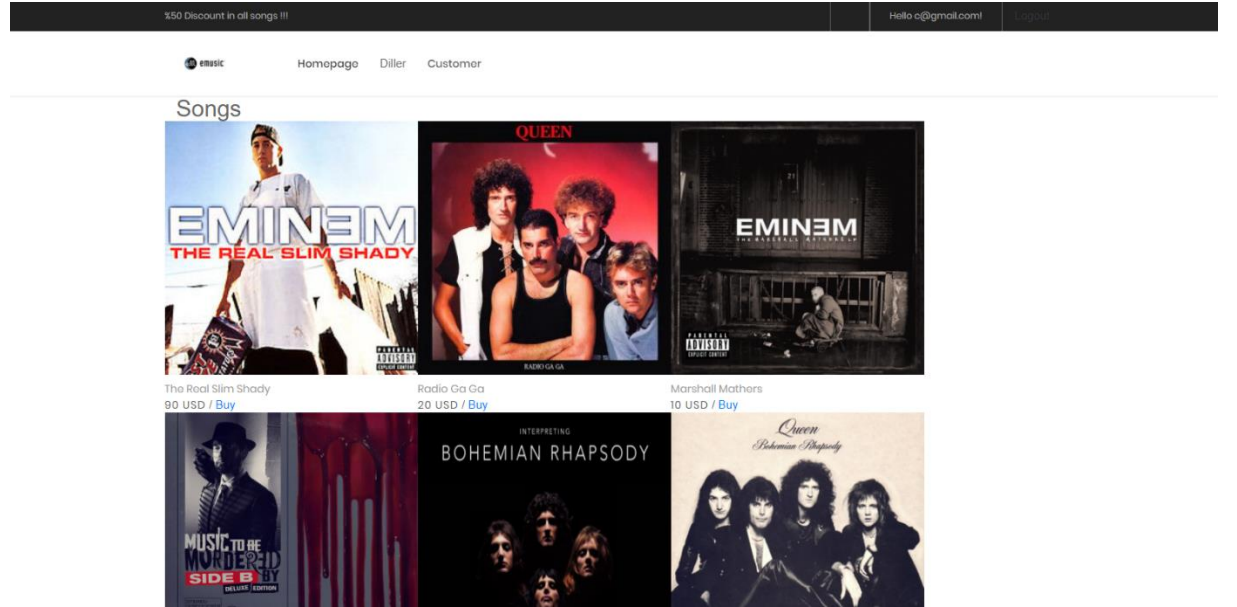
Veritabanımızda toplam 13 farklı tablo bulunmaktadır. Song tablosunda uygulamanın ana yapısını oluşturan ürün olan şarkıların detayları bulunmaktadır. Bu tabloda ArtistId, Artist tablosuyla ; CategoryId de Category tablosu ile ilişkilendirilmiştir. Artist

tablosunda şarkıların sahibi olan sanatçıların isimleri ve aktiflik durumu vardır. Category tablosunda şarkının içine girdiği kategori, adı ve aktiflik durumu vardır. Basket tablosu uygulamamızdaki alışveriş sepeti kısmıyla ilgilidir. Sepeti oluşturan kullanıcı, sepet içindeki şarkılar, fiyat aktiflik durumu ve sepetin durumu gibi özellikler tutulur. BasketItem tablosu içinde ise Basket tablosu içindeki her bir şarkının özellikleri tutulur. Şarkının hangi sepete ait olduğu, şarkı id'si ve aktiflik gibi özellikler tutulur. Discount tablosunda eğer istenirse indirim ekleme gerçekleştirilebilir. Geri kalan tablolar ASP NET CORE'un Identity ile otomatik olarak oluşturduğu tablolardır.

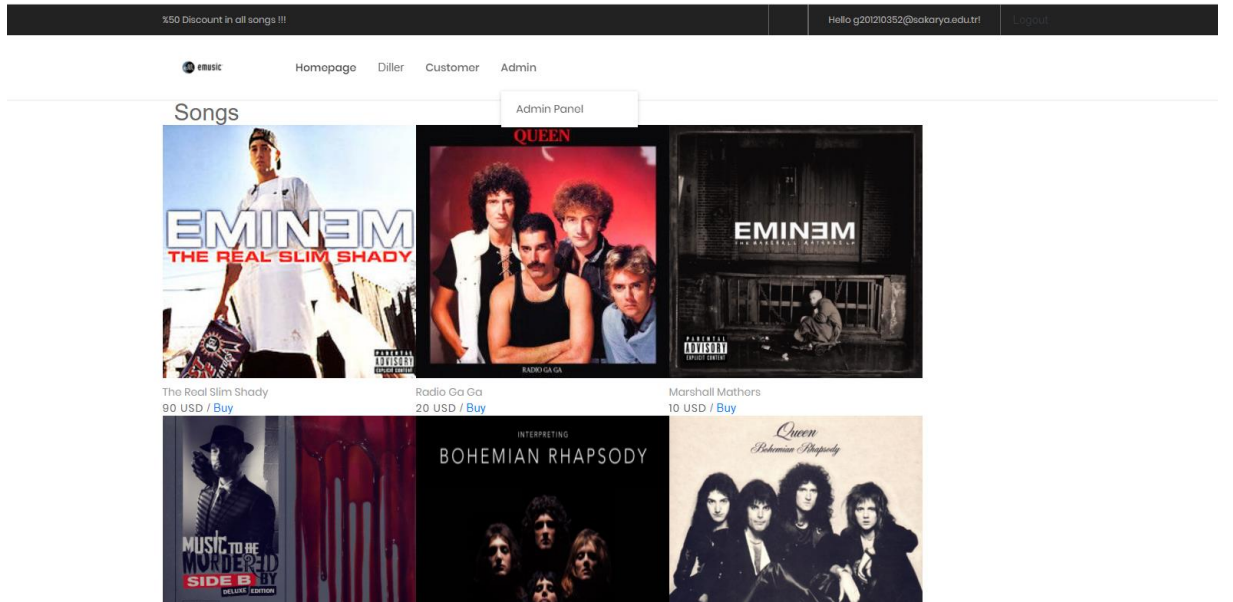
EKRAN GÖRÜNTÜLERİ:

Rol Yönetimi:

Müşteri giriş yapınca:



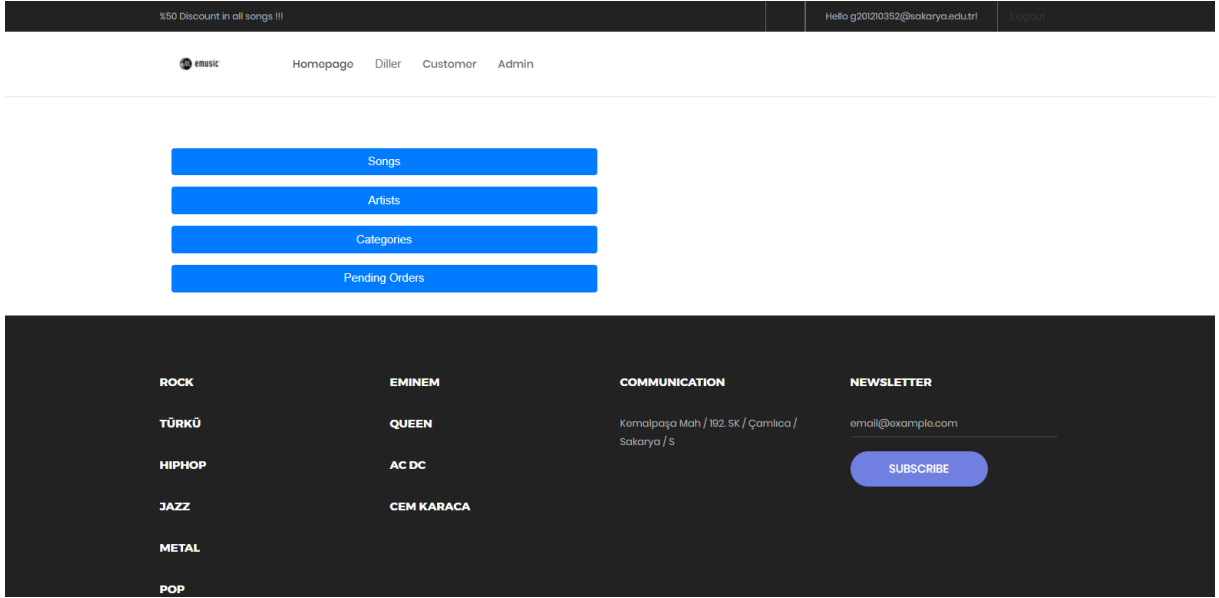
Admin giriş yapınca:



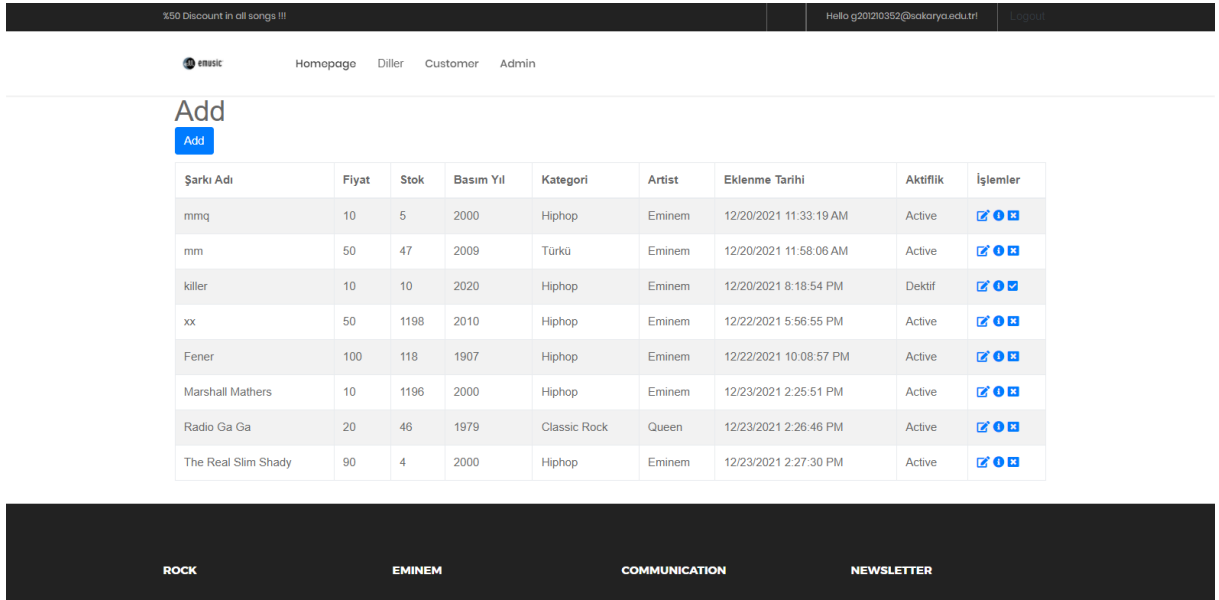
Admin Paneli:

Şarkılar, artistler ve kategoriler bölümleri(temel yapıları aynı olduğu için sadece şarkı'yı atıyorum):

Ana Ekran



Şarkı Ana Sayfa



Şarkı ekle

%50 Discount in all songs !!!

Hello g20120352@osakarya.edu.tr

Logout

EMUSIC

Homepage Diller Customer Admin

Add

Şarkı Adı

Kategori

Hiphop

Açıklama

Artist

Eminem

Fiyat

Birinci Fotoğraf

Dosya Seç Dosya seçilmedi

Stok

İkinci Fotoğraf

Dosya Seç Dosya seçilmedi

Basım Yılı

Üçüncü Fotoğraf

Dosya Seç Dosya seçilmedi

Add

Go back to list

Şarkı düzenle

%50 Discount in all songs !!!

Hello g20120352@osakarya.edu.tr

Logout

EMUSIC

Homepage Diller Customer Admin

EMusic

Şarkı Adı

The Real Slim Shady

Açıklama

Best single of 2000

Fiyat

90

Stok

4

Basım Yılı

2000

Kategori

Hiphop

Artist


Eminem

Ekleme Tarihi


12/23/2021 2:27:30 PM

Aktiflik


Birinci Fotoğraf



İkinci Fotoğraf




Üçüncü Fotoğraf



Edit

Go back to list

Şarkı aktiflik-deaktiflik

 [Homepage](#) [Diller](#) [Customer](#) [Admin](#)

Deaktifleştire

Şarkı Adı

mmq

Açıklama

mm

Fiyat

10

Stok

5

Basım Yılı

2000

Kategori

Hiphop


Artist

Eminem


Ekleme Tarihi

12/20/2021 11:33:19 AM


Birinci Fotoğraf



İkinci Fotoğraf



Üçüncü Fotoğraf

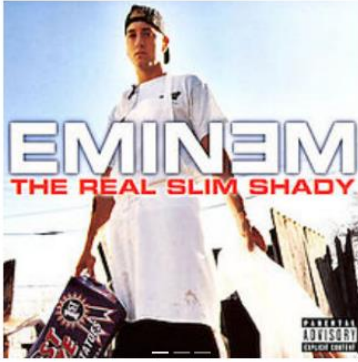


Deactive

Go back to list

Müşteri Şarkıyı Sepete Eklerken:

The Real Slim Shady



Best single of 2000

Stok: 4


Fiyat: 100 TL

Discount Price: 90 USD


Profit: 90 USD (100%)

Add To Basket


Other Songs




XX
50 TL



Marshall Mathers
10 TL



Fener
100 TL



killer
10 TL

Sepet:



Radio
Ga Ga
20 TL /
Sepetten
Kaldır



The
Real
Slim
Shady
90 TL /
Sepetten
Kaldır

TotalAmount110 Go To Payment

Ödeme İşlemleri:

150 discount on all songs !!!

hello.guzelsoz@sakarya.edu.tr

0000000000000000

etmusic

Homepage Diller Customer Admin

Pay with your Card

Pay with Card

ROCK

TÜRKÜ

HIPHOP

JAZZ

METAL

POP

EMINEM

QUEEN

AC DC

CEM KARACA

COMMUNICATION

Kemalpaşa Mah / 192. SK / Çamlıca /
Sakarya / S

NEWSLETTER

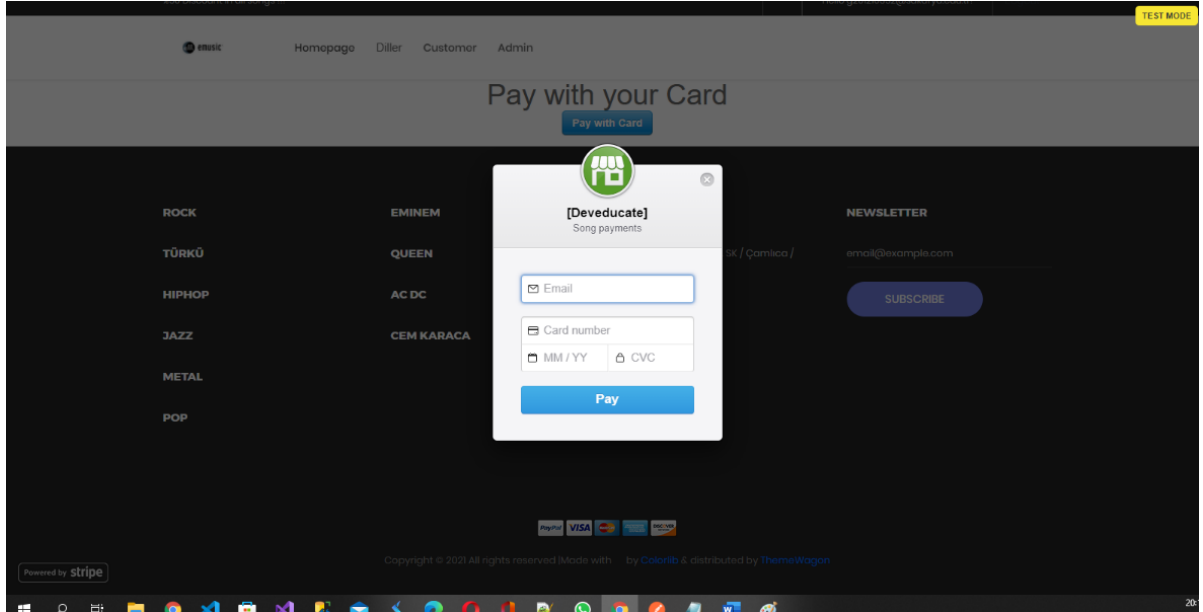
email@example.com

SUBSCRIBE

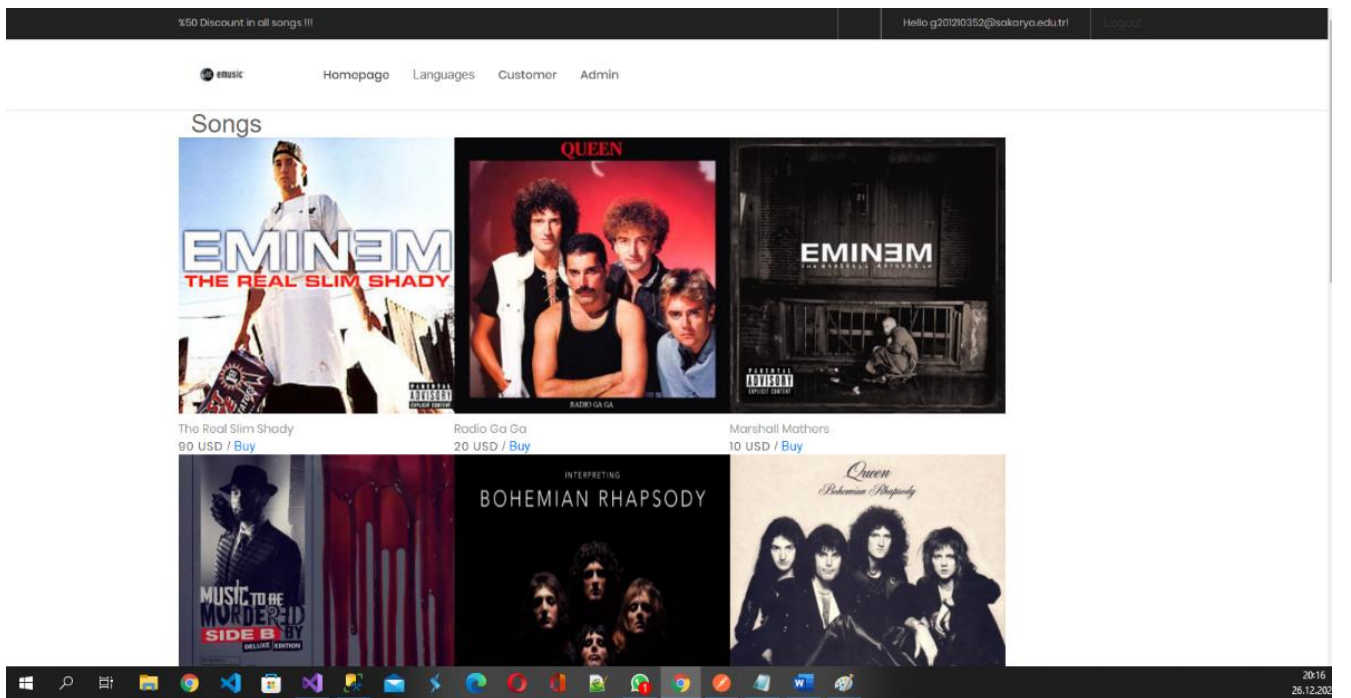
Paycom VISA Mastercard Apple Pay Google Pay

Copyright © 2021 All rights reserved | Made with by Colorlib & distributed by ThemeWagon

Stripe API ile demo kart bilgileri girilebilir ve alışveriş tamamlanır.



Çoklu Dil :



Şarkılar



The Real Slim Shady
90 TL / [Satın Al](#)



Radio Ga Ga
20 TL / [Satın Al](#)



Marshall Mathers
10 TL / [Satın Al](#)



API :

Api get işlemi:

GET https://localhost:44333/api/song/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "SongName": "mmq",
3   ... "Price": 10.0,
4   ... "Quantity": 0.0,
5   ... "PublishYear": 2000,
6   ... "ReleaseDate": 2022,
7   ... "Description": "mm",
8   ... "CreateDate": "2021-12-26T11:33:19.7433333",
9   ... "Stock": 10,
10  ... "Active": true
11 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 5 ms Size: 1.66 KB Save Response

Pretty Raw Preview Visualize JSON

```
70 {
71   "CreateDate": "2021-12-23T14:25:51.2420981",
72   "Stock": 1196,
73   "Active": true
74 },
75 {
76   "Id": 7,
77   "SongName": "Radio Ga Ga",
78   "Price": 20.0,
```

Api Post işlemi:

POST <https://localhost:44333/api/song/> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```
1 {
2   "SongName": "Apideneme123",
3   "Price": 10.0,
4   "Quantity": 0.0,
5   "PublishYear": 2000,
6   "ReleaseDate": 2022,
7   "Description": "api",
8   "CreateDate": "2021-12-29T11:33:19.7433333",
9   "Stock": 27,
10  "Active": true
11 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 854 ms Size: 194 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
1 "Added Successfully"
```

Api Put işlemi:

PUT <https://localhost:44333/api/song/> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```
1 {
2   "SongName": "Apideneme123",
3   "Price": 10.0,
4   "Quantity": 0.0,
5   "PublishYear": 2000,
6   "ReleaseDate": 2022,
7   "Description": "api",
8   "CreateDate": "2021-12-29T11:33:19.7433333",
9   "Stock": 29,
10  "Active": true
11 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 15 ms Size: 196 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
1 "Updated Successfully"
```

Continue learning Runner Trash Settings Help

API Delete işlemi:

The screenshot displays a REST client interface for a DELETE request. The URL bar shows `https://localhost:44333/api/song/11`. The request method is set to **DELETE**. The request body is in JSON format, containing the following data:

```
1 {
2   "SongName": "Apideneme123",
3   "Price": 10.0,
4   "Quantity": 0.0,
5   "PublishYear": 2000,
6   "ReleaseDate": 2022,
7   "Description": "api",
8   "CreateDate": "2021-12-29T11:33:19.7433333",
9   "Stock": 29,
10  "Active": true
11 }
```

The response status is **200 OK** with a time of 9 ms and a size of 196 B. The response body is in JSON format, showing:

```
1 "Deleted Successfully"
```

The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The Body tab is active, showing the request and response. The response is displayed in the 'Body' tab, showing the status '200 OK' and the message 'Deleted Successfully'.