

PROJE RAPORU

PROJE

RAPOR TARİHİ

PROJE ADI

HAZIRLAYAN:

26.12.2022

Online Müzik Uygulaması

Yusuf Türken

PROJE ÖZETİ

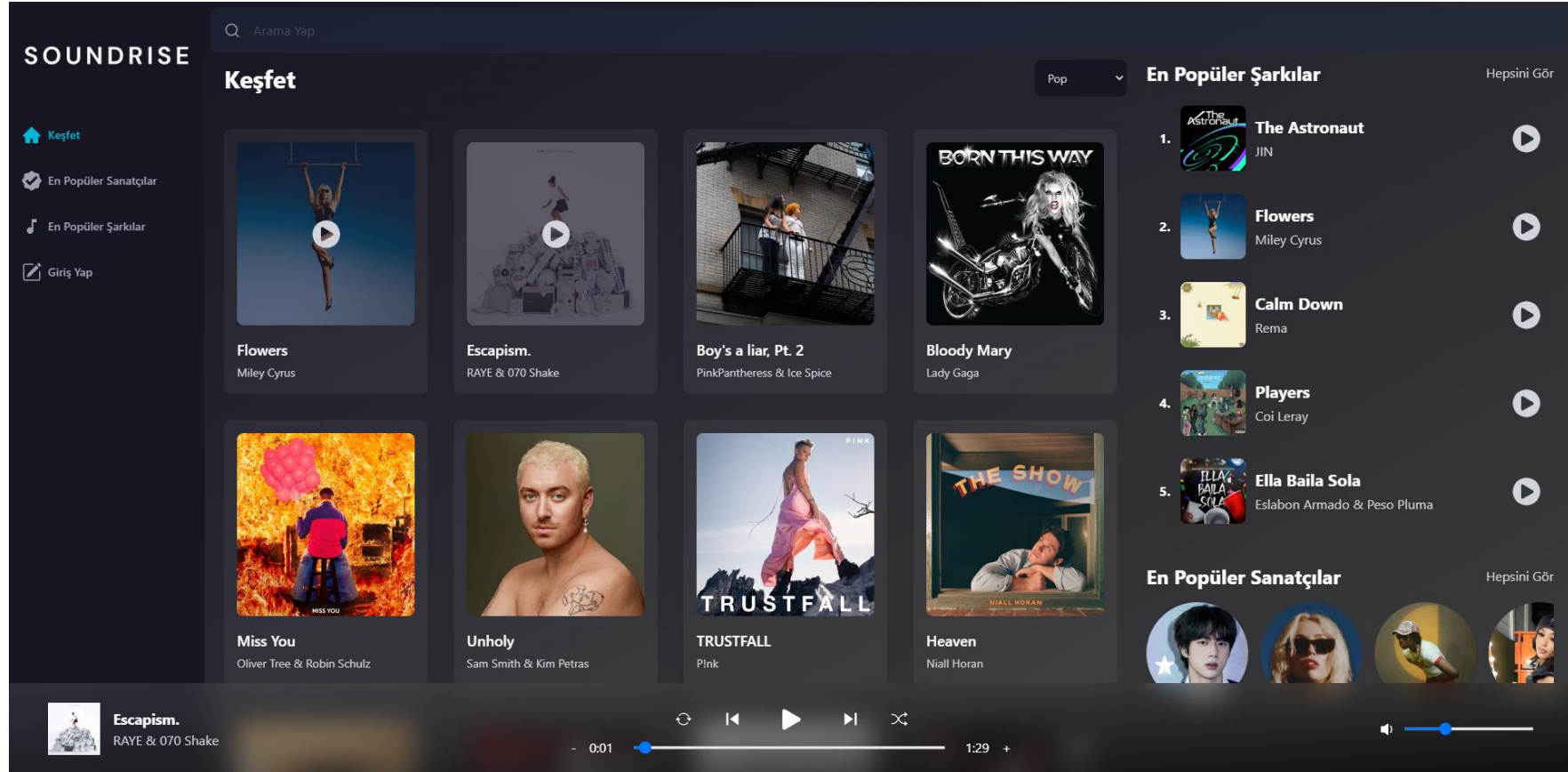
Hazırlanmış olan Online Müzik Uygulaması; Keşfet, En Popüler Sanatçılar, En Popüler Şarkılar, Şarkı Detayları, Sanatçı Detayları, Giriş Yap, Müzik Arama bölümleri bulunmaktadır. Yusuf Türken tarafından 21 MART-24 MART arasında geliştirilmiştir.

GÖREV	YAPILAN İŞLEMLER
Kullanılan Yazılım Dilleri ve Kütüphaneler	Yazılım dilleri HTML, CSS ve JavaScript'ten oluşmaktadır. Kullanılan kütüphaneler ise React JS, Redux Toolkit, Tailwind'dir.
Kullanılan API	Müzik, Müzik ismi, Müzik resmi gibi verileri çekmek için Rapid API da bulunan Shazam Core API tercih edilmiştir.
Sayfalar	Keşfet(Discover), Sanatçı Detayları(Artist Details), Müzik Detayları(Song Details), Müzik Arama(Search), En Popüler Sanatçılar(Top Artist), En Popüler Müzikler(Top Charts), Giriş Yap(Login).
Sayfa İçinde Kullanılan Bileşenler	Music Player, Artist Card, Details Header, Error, Loader, PlayPause, Related Songs, Search Bar, Side Bar, Song Bar, Song Card, Top Play. Yazılmış olan bu bileşenlerin açıklaması ve kodları bileşenler kısmında belirtilmiştir.
Araştırma	Uygulamada en çok araştırma API bulmak için yapılmıştır. Herhangi bir sorun ile karşılaşıldığında stackoverflow'dan yararlanılmıştır.

PROJE ZAMAN ÇİZELGESİ

	1.Gün	2.Gün	3.Gün
Planlama			
Hazırlık			
Araştırma			
Tasarım			
Sayfalar ve Bileşenler			
Hata Ayıklama			
Rapor Hazırlama			

Sayfalar Görünüm (Pages)



Anasayfa (Discover)

Bu sayfa sitenin açılış sayfasıdır varsayılan olarak gelir. Bu sayfada varsayılan olarak beliren yerler sidebar, arama yap, müzik oynatıcısı, ve sağda yer alan en popüler şarkılar ve en popüler sanatçılar kısımlarına bileşenler kısmında bahsedilecektir. Orta bölümdeki keşfetin sağında bulunan Pop yazan bölümden istediği müzik türünü seçerek o müzik türünden olan müzikleri listeleyebilir veya dilerse müziklerin üstüne tıklayarak müziği oynatabilir.

SOUNDRISE

Q Sabaton

Arama sonucu gösteriliyor Sabaton

Keşfet

En Popüler Sanatçılar

En Popüler Şarkılar

Giriş Yap

The Last Stand
Sabaton

Panzerkampf
Sabaton

To Hell and Back
Sabaton

The Attack Of The Dead ...
Sabaton

Night Witches
Sabaton

Bismarck
Sabaton

Winged Hussars
Sabaton

Soldier of Heaven
Sabaton

En Popüler Şarkılar
Hepsini Gör

1. **The Astronaut**
JIN
2. **Flowers**
Miley Cyrus
3. **Calm Down**
Rema
4. **Players**
Coi Leray
5. **Ella Baila Sola**
Eslabon Armado & Peso Pluma

En Popüler Sanatçılar
Hepsini Gör

Calm Down
Rema

0:00 1:29 +

Arama (Search)

Sayfanın en üst orta bölümünde bulunan arama çubuğundan ilgili artist, müzik veya müzik türü gibi istenildiği türde bir arama yapılabilir. Arama sonuçları hemen aşağıda listelenecektir.

SOUNDRISE

Arama Yap

Keşfet

En Popüler Sanatçılar

En Popüler Şarkılar

Giriş Yap

Metro Boomin
Hip-Hop/Rap

En İyi Şarkıları:

1. **Ric Flair Drip**
Without Warning

2. **X (feat. Future)**
Savage Mode

3. **No Heart**
Savage Mode

En Popüler Şarkılar

Hepsini Gör

1. **The Astronaut**
JIN

2. **Flowers**
Miley Cyrus

3. **Calm Down**
Rema

4. **Players**
Coi Leray

5. **Ella Baila Sola**
Eslabon Armado & Peso Pluma

En Popüler Sanatçılar

Hepsini Gör

Calm Down
Rema


0:00 1:29


Sanatçı Sayfası (ArtistDetail)


Bu sayfaya en popüler sanatçılar sayfasından veya şarkıların üzerindeki sanatçı ismine tıklayıp ulaşılabilir. Sanatçının en iyi şarkıları ve sanatçı hakkında detaylar görüntülenebilir.


SOUNDRISE


Arama Yap

 Keşfet

 En Popüler Sanatçılar

 En Popüler Şarkılar

 Giriş Yap




Escapism.

RAYE & 070 Shake

Pop






Sözler:

Sleazin' and teasin', I'm sittin' on him
All of my diamonds are drippin' on him
I met him at the bar, it was 12 or somethin'
I ordered two more wines, 'cause tonight, I want him
A little context if you care to listen
I find myself in a shit position
The man that I love sat me down last night
And he told me that it's over, dumb decision
And I don't wanna feel how my heart is rippin'
In fact, I don't wanna feel, so I stick to sippin'
And I'm out on the town with a simple mission
In my little black dress, and this shit is sittin'




Calm Down

Rema






- 0:00

1:29 +







En Popüler Şarkılar

Hepsini Gör

-  **The Astronaut**
JIN
-  **Flowers**
Miley Cyrus
-  **Calm Down**
Rema
-  **Players**
Coi Leray
-  **Ella Baila Sola**
Eslabon Armado & Peso Pluma

En Popüler Sanatçılar

Hepsini Gör

Şarkı Detayları (Song Details)


Bu sayfaya şarkı ismine tıklanılarak gidilebilir. Gelen sayfada şarkının ismi, sanatçının ismi, şarkı türü, sözler ve en aşağısında bu şarkıyla benzer türdeki şarkılar listelenir.

SOUNDRISE


[Keşfet](#)[En Popüler Sanatçılar](#)[En Popüler Şarkılar](#)[Giriş Yap](#)

Arama Yap


En Popüler Sanatçılar




JIN




Miley Cyrus




Rema




Coi Leray




Eslabon Armado & Peso ...



RAYE & 070 Shake




Ayra Starr



DNDM


En Popüler Şarkılar

Hepsini Gör

1. 


The Astronaut

JIN

2. 


Flowers

Miley Cyrus

3. 


Calm Down

Rema

4. 

Players

Coi Leray





5. 


Ella Baila Sola

Eslabon Armado & Peso Pluma

En Popüler Sanatçılar

Hepsini Gör





Calm Down

Rema

0:00

1:29

+

En Popüler Sanatçılar (TopArtists)


Bu sayfa şu anda en popüler olan sanatçılar listeler. Bu sayfaya sayfanın sol tarafında bulunan sidebar bölümünden En Popüler Sanatçılar seçeneğine tıklanılarak görüntülenebilir.

SOUNDRISE


[Keşfet](#)
[En Popüler Sanatçılar](#)
[En Popüler Şarkılar](#)
[Giriş Yap](#)

Arama Yap


En Popüler Şarkılar




The Astronaut
JIN




Flowers
Miley Cyrus




Calm Down
Rema




Players
Coi Leray




Ella Baila Sola
Eslabon Armado & Peso Pluma



Escapism.
RAYE & 070 Shake




Rush
Ayra Starr





Dubai (Hussein Arbabi R...)
DNDM


En Popüler Şarkılar


Hepsini Gör

1. **The Astronaut**
JIN

2. **Flowers**
Miley Cyrus





3. **Calm Down**
Rema


4. **Players**
Coi Leray

5. **Ella Baila Sola**
Eslabon Armado & Peso Pluma

En Popüler Sanatçılar

Hepsini Gör





Calm Down
Rema

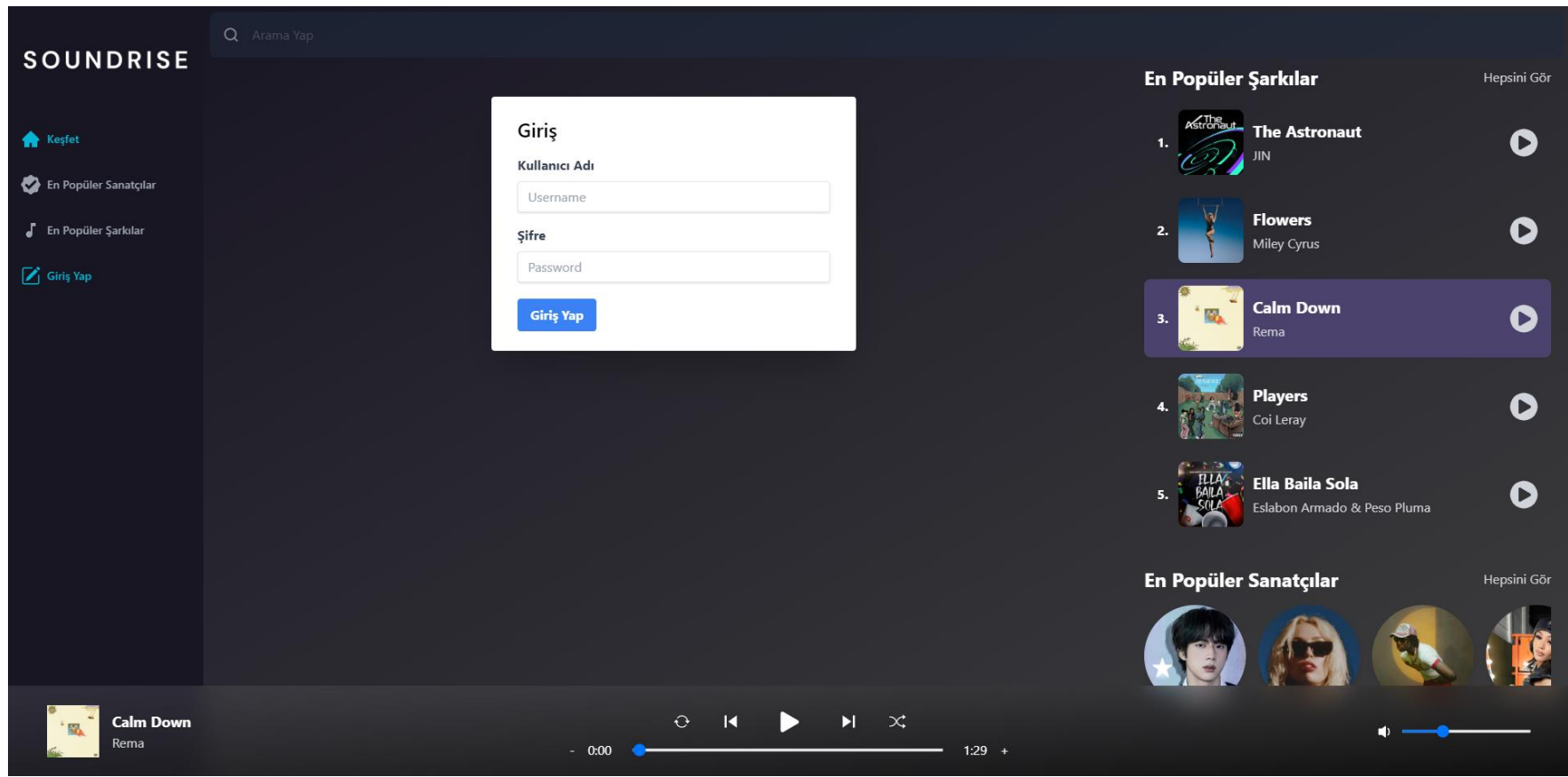
0:00

1:29

+

En Popüler Şarkılar (TopCharts)

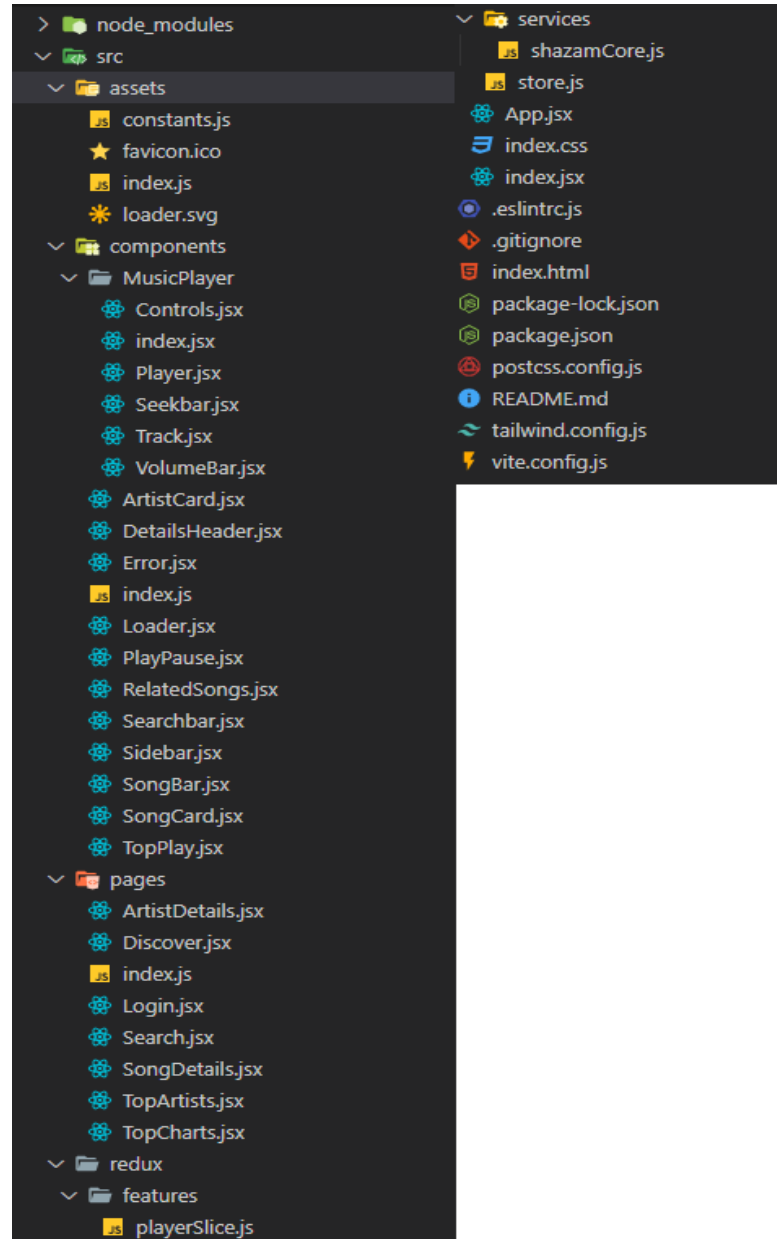
Bu sayfa şu anda en popüler olan şarkıları sırası ile listeler. Bu sayfaya ulaşmak için sayfanın sol tarafında bulunan En Popüler Şarkılar seçeneğine tıklanarak gidilebilir.



Giriş (Login)

Bu sayfanın kullanıcının giriş işlemini yapması için oluşturulmuştur. Girilen değerler localstorage’da geçici olarak depolanır ve gerekli işlemler için kullanılabilir.

Dosya Yapısı (File Structer)



Bileşenler (Components)

Müzik Oynatıcısı (Music Player)

Müzik oynatıcısı bir müziğin üzerine tıklanıldığında açılır ve aşağıdan bir müzik oynatıcı gelir.

```
import React from 'react';
import { MdSkipNext, MdSkipPrevious } from 'react-icons/md';
import { BsArrowRepeat, BsFillPauseFill, BsFillPlayFill, BsShuffle } from 'react-icons/bs';

const Controls = ({ isPlaying, repeat, setRepeat, shuffle, setShuffle, currentSongs, handlePlayPause,
handlePrevSong, handleNextSong }) => (
  <div className="flex items-center justify-around md:w-36 lg:w-52 2xl:w-80">
    <BsArrowRepeat size={20} color={repeat ? 'red' : 'white'} onClick={() => setRepeat((prev) => !prev)}
className="hidden sm:block cursor-pointer" />
    {currentSongs?.length && <MdSkipPrevious size={30} color="#FFF" className="cursor-pointer"
onClick={handlePrevSong} />}
    {isPlaying ? (
      <BsFillPauseFill size={45} color="#FFF" onClick={handlePlayPause} className="cursor-pointer" />
    ) : (
      <BsFillPlayFill size={45} color="#FFF" onClick={handlePlayPause} className="cursor-pointer" />
    )}
    {currentSongs?.length && <MdSkipNext size={30} color="#FFF" className="cursor-pointer"
onClick={handleNextSong} />}
    <BsShuffle size={20} color={shuffle ? 'red' : 'white'} onClick={() => setShuffle((prev) => !prev)}
className="hidden sm:block cursor-pointer" />
  </div>
);

export default Controls;
```

Controls

Bu sayfadaki kodlar müzik oynatıcısının; müzikleri durdurup başlatma, müzikler arası ileri geri, müzik tekrarlanması, rastgele müzik açılması gibi kontroller burada yazılmıştır.

```

import React, { useRef, useEffect } from 'react';

const Player = ({ activeSong, isPlaying, volume, seekTime, onEnded, onTimeUpdate, onLoadedData, repeat }) => {
  const ref = useRef(null);
  if (ref.current) {
    if (isPlaying) {
      ref.current.play();
    } else {
      ref.current.pause();
    }
  }

  useEffect(() => {
    ref.current.volume = volume;
  }, [volume]);
  useEffect(() => {
    ref.current.currentTime = seekTime;
  }, [seekTime]);

  return (
    <audio
      src={activeSong?.hub?.actions[1]?.uri}
      ref={ref}
      loop={repeat}
      onEnded={onEnded}
      onTimeUpdate={onTimeUpdate}
      onLoadedData={onLoadedData}
    />
  );
};

export default Player;

```

Player

```

import React from 'react';

const Seekbar = ({ value, min, max, onInput, setSeekTime, appTime }) => {

  const getTime = (time) => `${Math.floor(time / 60)}:${Math.floor(time % 60)}.slice(-2)`.slice(-2);

  return (
    <div className="hidden sm:flex flex-row items-center">
      <button type="button" onClick={() => setSeekTime(appTime - 5)} className="hidden lg:mr-4 lg:block text-white">
        -
      </button>
      <p className="text-white">{value === 0 ? '0:00' : getTime(value)}</p>
      <input
        type="range"
        step="any"
        value={value}
        min={min}
        max={max}
        onInput={onInput}
        className="md:block w-24 md:w-56 2xl:w-96 h-1 mx-4 2xl:mx-6 rounded-lg"
      />
      <p className="text-white">{max === 0 ? '0:00' : getTime(max)}</p>
      <button type="button" onClick={() => setSeekTime(appTime + 5)} className="hidden lg:ml-4 lg:block text-white">
        +
      </button>
    </div>
  );
};

export default Seekbar;

```

Seekbar

Bu kod parçası müzik çaların zamanını ve zamanını 5 artıran veya 5 eksiltten kodu içerir

```
import React from 'react';

const Track = ({ isPlaying, isActive, activeSong }) => (
  <div className="flex-1 flex items-center justify-start">
    <div className={`hidden sm:block h-16 w-16 mr-4`}>
      <img src={activeSong?.images?.coverart} alt="cover art" className="" />
    </div>
    <div className="w-[50%]">
      <p className="truncate text-white font-bold text-lg">
        {activeSong?.title ? activeSong?.title : 'No active Song'}
      </p>
      <p className="truncate text-gray-300">
        {activeSong?.subtitle ? activeSong?.subtitle : 'No active Song'}
      </p>
    </div>
  </div>
);

export default Track;
```

Track

Bu kod müzik resmini, ismin, sanatçı bilgisini göstermek için kullanılır.


```

import React from 'react';
import { BsFillVolumeUpFill, BsVolumeDownFill, BsFillVolumeMuteFill } from 'react-icons/bs';

const VolumeBar = ({ value, min, max, onChange, setVolume }) => (
  <div className="hidden lg:flex flex-1 items-center justify-end">
    {value <= 1 && value > 0.5 && <BsFillVolumeUpFill size={25} color="#FFF" onClick={() => setVolume(0)} />}
    {value <= 0.5 && value > 0 && <BsVolumeDownFill size={25} color="#FFF" onClick={() => setVolume(0)} />}
    {value == 0 && <BsFillVolumeMuteFill size={25} color="#FFF" onClick={() => setVolume(1)} />}
    <input
      type="range"
      step="any"
      value={value}
      min={min}
      max={max}
      onChange={onChange}
      className="2xl:w-40 lg:w-32 md:w-32 h-1 ml-2"
    />
  </div>
);

export default VolumeBar;

```

VolumeBar

Bu kod parçası müziğin sesinin açılıp kapanmasını sağlayan inputu ekrana getirir bu sayede kullanıcı müziğin sesini dilediği gibi ayarlayabilir.

```
import React, { useState, useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux';

import { nextSong, prevSong, playPause } from '../../redux/features/playerSlice';
import Controls from './Controls';
import Player from './Player';
import Seekbar from './Seekbar';
import Track from './Track';
import VolumeBar from './VolumeBar';

const MusicPlayer = () => {
  const { activeSong, currentSongs, currentIndex, isActive, isPlaying } = useSelector((state) => state.player);
  const [duration, setDuration] = useState(0);
  const [seekTime, setSeekTime] = useState(0);
  const [appTime, setAppTime] = useState(0);
  const [volume, setVolume] = useState(0.3);
  const [repeat, setRepeat] = useState(false);
  const [shuffle, setShuffle] = useState(false);
  const dispatch = useDispatch();

  useEffect(() => {
    if (currentSongs.length) dispatch(playPause(true));
  }, [currentIndex]);

  const handlePlayPause = () => {
    if (!isActive) return;

    if (isPlaying) {
      dispatch(playPause(false));
    } else {
      dispatch(playPause(true));
    }
  };

  const handleNextSong = () => {
    dispatch(playPause(false));
  };
}
```

```

    if (!shuffle) {
      dispatch(nextSong((currentIndex + 1) % currentSongs.length));
    } else {
      dispatch(nextSong(Math.floor(Math.random() * currentSongs.length)));
    }
  };

const handlePrevSong = () => {
  if (currentIndex === 0) {
    dispatch(prevSong(currentSongs.length - 1));
  } else if (shuffle) {
    dispatch(prevSong(Math.floor(Math.random() * currentSongs.length)));
  } else {
    dispatch(prevSong(currentIndex - 1));
  }
};

return (
  <div className="relative sm:px-12 px-8 w-full flex items-center justify-between">
    <Track isPlaying={isPlaying} isActive={isActive} activeSong={activeSong} />
    <div className="flex-1 flex flex-col items-center justify-center">
      <Controls
        isPlaying={isPlaying}
        isActive={isActive}
        repeat={repeat}
        setRepeat={setRepeat}
        shuffle={shuffle}
        setShuffle={setShuffle}
        currentSongs={currentSongs}
        handlePlayPause={handlePlayPause}
        handlePrevSong={handlePrevSong}
        handleNextSong={handleNextSong}
      />
      <SeekBar
        value={appTime}

```

```

        min="0"
        max={duration}
        onInput={(event) => setSeekTime(event.target.value)}
        setSeekTime={setSeekTime}
        appTime={appTime}
      />
      <Player
        activeSong={activeSong}
        volume={volume}
        isPlaying={isPlaying}
        seekTime={seekTime}
        repeat={repeat}
        currentIndex={currentIndex}
        onEnded={handleNextSong}
        onTimeUpdate={(event) => setAppTime(event.target.currentTime)}
        onLoadedData={(event) => setDuration(event.target.duration)}
      />
    </div>
    <VolumeBar value={volume} min="0" max="1" onChange={(event) => setVolume(event.target.value)}
    setVolume={setVolume} />
  </div>
);
};

export default MusicPlayer;

```

Music Player index

Bu kodlarda music player için gerekli componentler birleştirilmiş ve bazı ayarlamalar yapılmıştır.

Main Bileşenler (Main Components)

```
import React from 'react';
import { useNavigate } from 'react-router-dom';

const ArtistCard = ({ track }) => {
  const navigate = useNavigate();

  return (
    <div
      className="flex flex-col w-[250px] p-4 bg-white/5 bg-opacity-80 backdrop-blur-sm animate-slideup rounded-lg
cursor-pointer"
      onClick={() => navigate(`/artists/${track?.artists[0].adamid}`)}
    >
      <img alt="song_img" src={track?.images?.coverart} className="w-full h-56 rounded-lg" />
      <p className="mt-4 font-semibold text-lg text-white truncate">
        {track?.subtitle}
      </p>
    </div>
  );
};

export default ArtistCard;
```

Artist Card

```
import React from 'react';
import { Link } from 'react-router-dom';

const DetailsHeader = ({ artistId, artistData, songData }) => (
  <div className="relative w-full flex flex-col">
    <div className="w-full sm:h-48 h-28" />

    <div className="absolute inset-0 flex items-center">
      <img
        alt="profile"
        src={
          artistId ? artistData?.attributes?.artwork?.url
            .replace('{w}', '500')
            .replace('{h}', '500')
          : songData?.images?.coverart
        }
        className="sm:w-48 w-28 sm:h-48 h-28 rounded-full object-cover border-2 shadow-xl shadow-black"
        />

      <div className="ml-5">
        <p className="font-bold sm:text-3xl text-xl text-white">
          {artistId ? artistData?.attributes?.name : songData?.title}
        </p>
        <{!artistId && (
          <Link to={` /artists/${songData?.artists[0]?.adamid}`}>
            <p className="text-base text-gray-400 mt-2">{songData?.subtitle}</p>
          </Link>
        )}>

        <p className="text-base text-gray-400 mt-2">
          {artistId
            ? artistData?.attributes?.genreNames[0]
            : songData?.genres?.primary}
        </p>
      </div>
    </div>
  </div>
);
```

```
    <div className="w-full sm:h-44 h-24" />  
  </div>  
);  
  
export default DetailsHeader;
```

Details Header

Bu kod Artist Details sayfasındaki en üstte bulunan sanatçının resmini, ismini gibi bilgileri görüntüler.


```
import React from 'react';

const Error = () => (
  <div className="w-full flex justify-center items-center">
    <h1 className="font-bold text-2xl text-white">Something went wrong. Please try again</h1>
  </div>
);

export default Error;
```

Error

Bu kod sayfada herhangi bir hata oluştuğunda gösterilir.

```
import React from 'react';

import { loader } from '../assets';

const Loader = ({ title }) => (
  <div className="w-full flex justify-center items-center flex-col">
    <img src={loader} alt="loader" className="w-32 h-32 object-contain" />
    <h1 className="font-bold text-2xl text-white mt-2">{title || 'Loading'}</h1>
  </div>
);

export default Loader;
```

Loader

Bu kod müzikler api'dan fetch edilirken bir animasyon çıkarır.

```
import React from 'react';
import { FaPauseCircle, FaPlayCircle } from 'react-icons/fa';

const PlayPause = ({ isPlaying, activeSong, song, handlePause, handlePlay }) => (isPlaying && activeSong?.title
=== song.title ? (
  <FaPauseCircle
    size={35}
    className="text-gray-300"
    onClick={handlePause}
  />
) : (
  <FaPlayCircle
    size={35}
    className="text-gray-300"
    onClick={handlePlay}
  />
));

export default PlayPause;
```

PlayPause

Bu kod herhangi bir noktada play ve pause tuşlarını gerekli yerlerde daha sonra kullanılması için yazılmıştır.

```

import React from 'react';

import SongBar from './SongBar';

const RelatedSongs = ({ data, artistId, isPlaying, activeSong, handlePauseClick, handlePlayClick }) => {

  return (
    <div className="flex flex-col">
      <h1 className="font-bold text-3xl text-white">En İyi Şarkıları:</h1>

      <div className="mt-6 w-full flex flex-col">
        {data?.map((song, i) => (
          <SongBar
            key={` ${artistId}-${song.key}-${i}`}
            song={song}
            i={i}
            artistId={artistId}
            isPlaying={isPlaying}
            activeSong={activeSong}
          />
        ))}
      </div>
    </div>
  )
}

export default RelatedSongs;

```

Related Songs

Bu kod o anki müzikle alakalı müzikleri getirmek için şarkı detayları sayfasında kullanılmıştır.

```

import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';

import { FiSearch } from 'react-icons/fi';

const Searchbar = () => {
  const navigate = useNavigate();
  const [searchTerm, setSearchTerm] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();

    navigate(`/search/${searchTerm}`);
  };

  return (
    <div className='sm:p-2'>
      <form onSubmit={handleSubmit} autoComplete="off" className=" text-gray-200 focus-within:text-gray-600
rounded-lg bg-slate-800 opacity-60 hover:opacity-90">
        <label htmlFor="search-field" className="sr-only">
          Search all files
        </label>
        <div className="flex flex-row justify-start items-center">
          <FiSearch aria-hidden="true" className="w-5 h-5 ml-4" />
          <input
            name="search-field"
            autoComplete="off"
            id="search-field"
            className="flex-1 bg-transparent border-none placeholder-gray-500 outline-none text-base text-white p-
4"
            placeholder="Arama Yap"
            type="search"
            value={searchTerm}
            onChange={(e) => setSearchTerm(e.target.value)}
          />
        </div>
      </form>
    </div>
  );
};

```

```
        </form>
      </div>
    );
  };
export default Searchbar;
```

Searchbar

```

import React, { useState } from 'react';
import { NavLink } from 'react-router-dom';
import { HiOutlineMenu } from 'react-icons/hi';
import { IoMdMusicalNote } from 'react-icons/io';
import { GoVerified } from 'react-icons/go';
import { RiCloseLine } from 'react-icons/ri';
import { AiFillHome } from 'react-icons/ai';
import { BsPencilSquare } from 'react-icons/bs';

const links = [
  { name: 'Keşfet', to: '/', icon: AiFillHome },
  { name: 'En Popüler Sanatçılar', to: '/top-artists', icon: GoVerified },
  { name: 'En Popüler Şarkılar', to: '/top-charts', icon: IoMdMusicalNote },
  { name: 'Giriş Yap', to: '/login', icon: BsPencilSquare },
];

const NavLinks = ({ handleClick }) => (
  <div className="mt-6">
    {links.map((item) => (
      <NavLink
        key={item.name}
        to={item.to}
        className="flex flex-row justify-start items-center my-8 text-sm font-medium text-gray-400 hover:text-cyan-400"
        onClick={() => handleClick && handleClick()}
      >
        <item.icon className="w-6 h-6 mr-2" />
        {item.name}
      </NavLink>
    ))}
  </div>
);

const Sidebar = () => {
  const [mobileMenuOpen, setMobileMenuOpen] = useState(false);

```

```

return (
  <>
    <div className="md:flex hidden flex-col w-[240px] py-10 px-4 bg-[#191624]">
      
      <NavLinks />
    </div>

    <div className="absolute md:hidden block top-6 right-3">
      {!mobileMenuOpen ? (
        <HiOutlineMenu className="w-6 h-6 mr-4 mt-20 text-white" onClick={() => setMobileMenuOpen(true)} />
      ) : (
        <RiCloseLine className="w-6 h-6 mr-4 mt-20 text-white" onClick={() => setMobileMenuOpen(false)} />
      )}
    </div>

    <div className={`absolute top-0 h-screen w-2/3 bg-gradient-to-tl from-white/10 to-[#483D8B] backdrop-blur-lg
z-10 p-6 md:hidden smooth-transition ${mobileMenuOpen ? 'left-0' : '-left-full'}`}>
      
      <NavLinks handleClick={() => setMobileMenuOpen(false)} />
    </div>
  </>
);
};

export default Sidebar;

```

Sidebar

Bu kod sol tarafta bulunan sidebarı görüntülemek için kullanılır.


```
import React from 'react';
import { Link } from 'react-router-dom';

import PlayPause from './PlayPause';

const SongBar = ({ song, i, artistId, isPlaying, activeSong, handlePauseClick, handlePlayClick }) => {
  return (
    <div className={`w-full flex flex-row items-center hover:bg-[#4c426e] ${activeSong?.title === song?.title ?
'bg-[#4c426e]' : 'bg-transparent'} py-2 p-4 rounded-lg cursor-pointer mb-2`} >
      <h3 className="font-bold text-base text-white mr-3">{i + 1}</h3>
      <div className="flex-1 flex flex-row justify-between items-center">
        <img
          className="w-20 h-20 rounded-lg"
          src={artistId ? song?.attributes?.artwork?.url.replace('{w}', '125').replace('{h}', '125') :
song?.images?.coverart}
          alt={song?.title}
        />
        <div className="flex-1 flex flex-col justify-center mx-3">
          {!artistId ? (
            <Link to={`/songs/${song.key}`} >
              <p className="text-xl font-bold text-white">
                {song?.title}
              </p>
            </Link>
          ) : (
            <p className="text-xl font-bold text-white">
              {song?.attributes?.name}
            </p>
          )}
          <p className="text-base text-gray-300 mt-1">
            {artistId ? song?.attributes?.albumName : song?.subtitle}
          </p>
        </div>
      </div>
    </div>
  );
};
```

```
      <PlayPause
        isPlaying={isPlaying}
        activeSong={activeSong}
        song={song}
        handlePauseClick={handlePauseClick}
        handlePlayClick={() => handlePlayClick(song, i)}
      />
    )
    : null}
  </div>
)
}

export default SongBar;
```

SongBar

```

import React from 'react';
import { Link } from 'react-router-dom';
import { useDispatch } from 'react-redux';

import PlayPause from './PlayPause';
import { playPause, setActiveSong } from '../redux/features/playerSlice';

const SongCard = ({ song, isPlaying, activeSong, data, i }) => {
  const dispatch = useDispatch();

  const handlePauseClick = () => {
    dispatch(playPause(false));
  };

  const handlePlayClick = () => {
    dispatch(setActiveSong({ song, data, i }));
    dispatch(playPause(true));
  };

  return (
    <div className="flex flex-col w-[250px] p-4 bg-white/5 bg-opacity-80 backdrop-blur-sm animate-slideup rounded-
lg cursor-pointer">
      <div className="relative w-full h-56 group">
        <div className={`absolute inset-0 justify-center items-center bg-black bg-opacity-50 group-hover:flex
${activeSong?.title === song.title ? 'flex bg-black bg-opacity-70 rounded-md' : 'hidden'}`>
          <PlayPause
            isPlaying={isPlaying}
            activeSong={activeSong}
            song={song}
            handlePause={handlePauseClick}
            handlePlay={handlePlayClick}
          />
        </div>
        <img alt="Can't show because of the apple policy" src={song.images?.coverart} className="w-full h-full
rounded-lg" />
      </div>
    </div>
  );
};

```

```
    <div className="mt-4 flex flex-col">
      <p className="font-semibold text-lg text-white truncate">
        <Link to={` /songs/${song?.key}`}>
          {song.title}
        </Link>
      </p>
      <p className="text-sm truncate text-gray-300 mt-1">
        <Link to={song.artists ? `/artists/${song?.artists[0]?.adamid}` : '/top-artists'}>
          {song.subtitle}
        </Link>
      </p>
    </div>
  </div>
);
};

export default SongCard;
```

Song Card

Bu kod sayfalardaki müziklerin gösterildiği card yapısını oluşturmak için yazılmıştır.

```

import React, { useEffect, useRef } from 'react';
import { Link } from 'react-router-dom';
import { useSelector, useDispatch } from 'react-redux';
import { Swiper, SwiperSlide } from 'swiper/react';
import { FreeMode } from 'swiper';

import PlayPause from './PlayPause';
import { playPause, setActiveSong } from '../redux/features/playerSlice';
import { useGetTopChartsQuery } from '../redux/services/shazamCore';

import 'swiper/css';
import 'swiper/css/free-mode';

const TopChartCard = ({ song, i, isPlaying, activeSong, handlePauseClick, handlePlayClick }) => (
  <div className={`w-full flex flex-row items-center hover:bg-[#4c426e] ${activeSong?.title === song?.title ? 'bg-[#4c426e]' : 'bg-transparent'} py-2 p-4 rounded-lg cursor-pointer mb-2`} >
    <h3 className="font-bold text-base text-white mr-3">{i + 1}</h3>
    <div className="flex-1 flex flex-row justify-between items-center">
      <img className="w-20 h-20 rounded-lg" src={song?.images?.coverart} alt={song?.title} />
      <div className="flex-1 flex flex-col justify-center mx-3">
        <Link to={` /songs/${song.key}`} >
          <p className="text-xl font-bold text-white">
            {song?.title}
          </p>
        </Link>
        <Link to={` /artists/${song?.artists[0].adamid}`} >
          <p className="text-base text-gray-300 mt-1">
            {song?.subtitle}
          </p>
        </Link>
      </div>
    </div>
    <PlayPause
      isPlaying={isPlaying}
      activeSong={activeSong}
      song={song}
    />
  </div>
);

```

```

        handlePause={handlePauseClick}
        handlePlay={handlePlayClick}
      />
    </div>
  );

const TopPlay = () => {
  const dispatch = useDispatch();
  const { activeSong, isPlaying } = useSelector((state) => state.player);
  const { data } = useGetTopChartsQuery();
  const divRef = useRef(null);

  useEffect(() => {
    divRef.current.scrollToView({ behavior: 'smooth' });
  });

  const topPlays = data?.slice(0, 5);

  const handlePauseClick = () => {
    dispatch(playPause(false));
  };

  const handlePlayClick = (song, i) => {
    dispatch(setActiveSong({ song, data, i }));
    dispatch(playPause(true));
  };

  return (
    <div ref={divRef} className="xl:ml-6 ml-0 xl:mb-0 mb-6 flex-1 xl:max-w-[500px] max-w-full flex flex-col">
      <div className="w-full flex flex-col">
        <div className="flex flex-row justify-between items-center">
          <h2 className="text-white font-bold text-2xl">En Popüler Şarkılar</h2>
          <Link to="/top-charts">
            <p className="text-gray-300 text-base cursor-pointer">Hepsini Gör</p>
          </Link>
        </div>

```

```

<div className="mt-4 flex flex-col gap-1">
  {topPlays?.map((song, i) => (
    <TopChartCard
      key={song.key}
      song={song}
      i={i}
      isPlaying={isPlaying}
      activeSong={activeSong}
      handlePauseClick={handlePauseClick}
      handlePlayClick={() => handlePlayClick(song, i)}
    />
  ))}
</div>
</div>

<div className="w-full flex flex-col mt-8">
  <div className="flex flex-row justify-between items-center">
    <h2 className="text-white font-bold text-2xl">En Popüler Sanatçılar</h2>
    <Link to="/top-artists">
      <p className="text-gray-300 text-base cursor-pointer">Hepsini Gör</p>
    </Link>
  </div>

  <Swiper
    slidesPerView="auto"
    spaceBetween={15}
    freeMode
    centeredSlides
    centeredSlidesBounds
    modules={[FreeMode]}
    className="mt-4"
  >
    {topPlays?.slice(0, 5).map((artist) => (
      <SwiperSlide
        key={artist?.key}

```



```
        style={{ width: '25%', height: 'auto' }}
        className="shadow-lg rounded-full animate-slideright"
      >
        <Link to={` /artists/${artist?.artists[0].adamid}`} >
          <img src={artist?.images?.background} alt="Name" className="rounded-full w-full object-cover" />
        </Link>
      </SwiperSlide>
    )})
  </Swiper>
</div>
</div>
);
};

export default TopPlay;
```

TopPlay

Bu kod sayfanın sağında bulunan en popüler şarkılar ve en popüler sanatçılar kısmını oluşturmak için yazılmıştır.

Sayfalar Kodlar (Pages)

Artist Details

```
import React from 'react';
import { useParams } from 'react-router-dom';
import { useSelector } from 'react-redux';
import { DetailsHeader, Error, Loader, RelatedSongs } from '../components';

import { useGetArtistDetailsQuery } from '../redux/services/shazamCore';

const ArtistDetails = () => {
  const { id: artistId } = useParams();
  const { activeSong, isPlaying } = useSelector((state) => state.player);
  const { data: artistData, isFetching: isFetchingArtistDetails, error } = useGetArtistDetailsQuery(artistId);

  if (isFetchingArtistDetails) return <Loader title="Loading artist details..." />;

  if (error) return <Error />;

  return (
    <div className="flex flex-col">
      <DetailsHeader
        artistId={artistId}
        artistData={artistData?.data[0]}
      />

      <RelatedSongs
        data={artistData?.data[0].views['top-songs']?.data}
        artistId={artistId}
        isPlaying={isPlaying}
        activeSong={activeSong}
      />
    </div>
  );
};
```

```
);  
};  
  
export default ArtistDetails;
```

Discover (Anasayfa)

```
import React from 'react';  
import { useDispatch, useSelector } from 'react-redux';  
  
import { Error, Loader, SongCard } from '../components';  
import { selectGenreListId } from '../redux/features/playerSlice';  
import { useGetSongsByGenreQuery } from '../redux/services/shazamCore';  
import { genres } from '../assets/constants';  
  
const Discover = () => {  
  const dispatch = useDispatch();  
  const { genreListId } = useSelector((state) => state.player);  
  const { activeSong, isPlaying } = useSelector((state) => state.player);  
  const { data, isFetching, error } = useGetSongsByGenreQuery(genreListId || 'POP');  
  
  if (isFetching) return <Loader title="Yükleniyor..." />;  
  
  if (error) return <Error />;  
  
  const genreTitle = genres.find(({ value }) => value === genreListId)?.title;  
  
  return (  
    <div className="flex flex-col">  
      <div className="w-full flex justify-between items-center sm:flex-row flex-col mb-10">  
        <h2 className="font-bold text-3xl text-white text-left">Keşfet {genreTitle}</h2>  
  
        <select  
          onChange={(e) => dispatch(selectGenreListId(e.target.value))}>
```

```

        value={genreListId || 'pop'}
        className="bg-black text-gray-300 p-3 text-sm rounded-lg outline-none sm:mt-0 mt-5"
      >
        {genres.map((genre) => <option key={genre.value} value={genre.value}>{genre.title}</option>)}
      </select>
    </div>

    <div className="flex flex-wrap sm:justify-start justify-center gap-8">
      {data?.map((song, i) => (
        <SongCard
          key={song.key}
          song={song}
          isPlaying={isPlaying}
          activeSong={activeSong}
          data={data}
          i={i}
        />
      ))}
    </div>
  </div>
);
};

export default Discover;

```

Login

```
import React from 'react'
import { useState } from 'react'

const Login = () => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = (event) => {
    event.preventDefault();
    const user = { username, password };
    localStorage.setItem('user', JSON.stringify(user));
  };

  return (

    <div className="flex justify-center items-center mt-10">
      <form onSubmit={handleLogin} className="w-full max-w-md bg-white shadow-2xl rounded px-8 py-6">
        <h2 className="text-2xl font-medium mb-4">Giriş</h2>
        <div className="mb-4">
          <label className="block text-gray-700 font-bold mb-2">
            Kullanıcı Adı
          </label>
          <input value={username} onChange={(e) => setUsername(e.target.value)} className="shadow
appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-
outline" id="username" type="text" placeholder="Username" />
        </div>
        <div className="mb-4">
          <label className="block text-gray-700 font-bold mb-2">
            Şifre
          </label>
```

```
        <input value={password} onChange={(e) => setPassword(e.target.value)} className="shadow
appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-
outline" id="password" type="password" placeholder="Password" />
      </div>
      <div className="flex items-center justify-between mt-5">
        <button className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded
focus:outline-none focus:shadow-outline" type="submit">
          Giriş Yap
        </button>
      </div>
    </form>
  </div>
)
}

export default Login
```

Search

```
import React from 'react';
import { useSelector } from 'react-redux';
import { useParams } from 'react-router-dom';

import { Error, Loader, SongCard } from '../components';
import { useGetSongsBySearchQuery } from '../redux/services/shazamCore';

const Search = () => {
  const { searchTerm } = useParams();
  const { activeSong, isPlaying } = useSelector((state) => state.player);
  const { data, isFetching, error } = useGetSongsBySearchQuery(searchTerm);

  const songs = data?.tracks?.hits.map((song) => song.track);

  if (isFetching) return <Loader title={`Aranıyor ${searchTerm}...`} />;

  if (error) return <Error />;

  return (
    <div className="flex flex-col">
      <h2 className="font-bold text-3xl text-white text-left mt-4 mb-10">Arama sonucu gösteriliyor <span
className="font-black">{searchTerm}</span></h2>

      <div className="flex flex-wrap sm:justify-start justify-center gap-8">
        {songs.map((song, i) => (
          <SongCard
            key={song.key}
            song={song}
            isPlaying={isPlaying}
            activeSong={activeSong}
            data={data}
            i={i}
          />
        ))}
      </div>
    </div>
  );
}
```

```

    </div>
  </div>
);
};

export default Search;

```

Song Details

```

import React from 'react';
import { useParams } from 'react-router-dom';
import { useSelector, useDispatch } from 'react-redux';
import { DetailsHeader, Error, Loader, RelatedSongs } from '../components';

import { setActiveSong, playPause } from '../redux/features/playerSlice';
import { useGetSongDetailsQuery, useGetSongRelatedQuery } from '../redux/services/shazamCore';

import { FaSadTear } from 'react-icons/fa'

const SongDetails = () => {
  const dispatch = useDispatch();
  const { songid, id: artistId } = useParams();
  const { activeSong, isPlaying } = useSelector((state) => state.player);

  const { data, isFetching: isFetchinRelatedSongs, error } = useGetSongRelatedQuery({ songid });
  const { data: songData, isFetching: isFetchingSongDetails } = useGetSongDetailsQuery({ songid });

  if (isFetchingSongDetails && isFetchinRelatedSongs) return <Loader title="Searching song details" />;

  console.log(songData);

  if (error) return <Error />;

  const handlePauseClick = () => {

```



```

    dispatch(playPause(false));
  };

  const handlePlayClick = (song, i) => {
    dispatch(setActiveSong({ song, data, i }));
    dispatch(playPause(true));
  };

  return (
    <div className="flex flex-col">
      <DetailsHeader
        artistId={artistId}
        songData={songData}
      />

      <div className="mb-10">
        <h2 className="text-white text-3xl font-bold">Sözler:</h2>

        <div className="mt-5">
          {songData?.sections[1].type === 'LYRICS'
            ? songData?.sections[1]?.text.map((line, i) => (
                <p key={`lyrics-${line}-${i}`} className="text-gray-400 text-base my-1">{line}</p>
              ))
            : (
                <p className="text-gray-400 text-base my-1">Üzgünüz, herhangi bir şarkı sözü bulunmamaktadır!
<FaSadTear /></p>
              )}
        </div>
      </div>

      <RelatedSongs
        data={data}
        artistId={artistId}
        isPlaying={isPlaying}
        activeSong={activeSong}
        handlePauseClick={handlePauseClick}

```

```

        handlePlayClick={handlePlayClick}
      />

    </div>
  );
};

export default SongDetails;

```

Top Artist

```

import React from 'react';

import { ArtistCard, Error, Loader } from '../components';
import { useGetTopChartsQuery } from '../redux/services/shazamCore';

const TopArtists = () => {
  const { data, isFetching, error } = useGetTopChartsQuery();

  if (isFetching) return <Loader title="Loading artists..." />;

  if (error) return <Error />;

  return (
    <div className="flex flex-col">
      <h2 className="font-bold text-3xl text-white text-left mb-10">En Popüler Sanatçılar</h2>

      <div className="flex flex-wrap sm:justify-start justify-center gap-8">
        {data?.map((track) => <ArtistCard key={track.key} track={track} />)}
      </div>
    </div>
  );
};

```

```
export default TopArtists;
```

Top Charts

```
import React from 'react';
import { useSelector } from 'react-redux';

import { Error, Loader, SongCard } from '../components';
import { useGetTopChartsQuery } from '../redux/services/shazamCore';

const TopCharts = () => {
  const { data, isFetching, error } = useGetTopChartsQuery();
  const { activeSong, isPlaying } = useSelector((state) => state.player);

  if (isFetching) return <Loader title="Loading Top Charts" />;

  if (error) return <Error />;

  return (
    <div className="flex flex-col">
      <h2 className="font-bold text-3xl text-white text-left mb-10">En Popüler Şarkılar</h2>

      <div className="flex flex-wrap sm:justify-start justify-center gap-8">
        {data.map((song, i) => (
          <SongCard
            key={song.key}
            song={song}
            isPlaying={isPlaying}
            activeSong={activeSong}
            data={data}
            i={i}
          />
        ))}
      </div>
    </div>
  )
}
```

```

    </div>
  </div>
);
};

export default TopCharts;

```

Redux ve API

```

import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react';

export const api_key = '2928765eafmsh4812dbbbe4e0b32p1ae426jsnad7a062a5ce3'; // My api key which i get on Rapid API

export const shazamCoreApi = createApi({

  reducerPath: 'shazamCoreApi',
  baseQuery: fetchBaseQuery({
    baseUrl: 'https://shazam-core.p.rapidapi.com/',
    prepareHeaders: (headers) => {
      headers.set('X-RapidAPI-Key', api_key);

      return headers;
    },
  }),
  endpoints: (builder) => ({
    getTopCharts: builder.query({ query: () => 'v1/charts/world' }), // Top Musics
    getSongsByGenre: builder.query({ query: (genre) => `v1/charts/genre-world?genre_code=${genre}` }), // Fetching
    songs by genre
    getSongsBySearch: builder.query({ query: (searchTerm) =>
`v1/search/multi?search_type=SONGS_ARTISTS&query=${searchTerm}` }), // Search

```

```
    getArtistDetails: builder.query({ query: (artistId) => `v2/artists/details?artist_id=${artistId}`
  })),//Fetching artist detail with id
    getSongDetails: builder.query({ query: ({ songid }) => `v1/tracks/details?track_id=${songid}`  })),//Fetching
song details with
    getSongRelated: builder.query({ query: ({ songid }) => `v1/tracks/related?track_id=${songid}`  })),//Fetching
songs by genre
  })),
});

export const { //Exporting all things above
  useGetTopChartsQuery,
  useGetSongsByGenreQuery,
  useGetSongsBySearchQuery,
  useGetArtistDetailsQuery,
  useGetSongDetailsQuery,
  useGetSongRelatedQuery,
} = shazamCoreApi;
```

Shazam Core

Bu kod api'dan bütün dataları çekip export ediyor.

```
import { configureStore } from '@reduxjs/toolkit';

import { shazamCoreApi } from '../services/shazamCore';
import playerReducer from '../features/playerSlice';

export const store = configureStore({
  reducer: {
    [shazamCoreApi.reducerPath]: shazamCoreApi.reducer,
    player: playerReducer,
  },
  middleware: (getDefaultMiddleware) => getDefaultMiddleware().concat(shazamCoreApi.middleware),
});
```

Store

```
import { useSelector } from 'react-redux';
import { Route, Routes } from 'react-router-dom';

import { Searchbar, Sidebar, MusicPlayer, TopPlay } from '../components';
import { ArtistDetails, TopArtists, Discover, Search, SongDetails, TopCharts, Login } from '../pages';

const App = () => {
  const { activeSong } = useSelector((state) => state.player);

  return (
    <div className="relative flex">
      <Sidebar />
      <div className="flex-1 flex flex-col bg-gradient-to-br from-black to-[#3d3d3d]">
        <Searchbar />

        <div className="px-6 h-[calc(100vh-72px)] overflow-y-scroll hide-scrollbar flex xl:flex-row flex-col-reverse">
          <div className="flex-1 h-fit pb-40">
            <Routes>
```

```

        <Route path="/" element={<Discover />} />
        <Route path="/top-artists" element={<TopArtists />} />
        <Route path="/top-charts" element={<TopCharts />} />
        <Route path="/artists/:id" element={<ArtistDetails />} />
        <Route path="/songs/:songid" element={<SongDetails />} />
        <Route path="/search/:searchTerm" element={<Search />} />
        <Route path="/login" element={<Login />} />
      </Routes>
    </div>
    <div className="xl:sticky relative top-0 h-fit">
      <TopPlay />
    </div>
  </div>
</div>

    {activeSong?.title && (
      <div className="absolute h-28 bottom-0 left-0 right-0 flex animate-slideup bg-gradient-to-br from-white/10
to-[#0f0f0f] backdrop-blur-lg z-10">
        <MusicPlayer />
      </div>
    )}
  </div>
);
};

export default App;

```

App

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter as Router } from 'react-router-dom';
import { Provider } from 'react-redux';

import './index.css';
import App from './App';
import { store } from './redux/store';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <Provider store={store}>
      <Router>
        <App />
      </Router>
    </Provider>
  </React.StrictMode>,
);
```

index

Kurulum

1. Proje klasörü vscode da açın.
2. Terminale “npm install” yazın. Yükleme tamamlandıktan sonra diğer adıma geçin.
3. Terminale “npm start” yazın.
4. Browser’da search bar’a localhost:3000 yazarak projeye ulaşabilirsiniz.