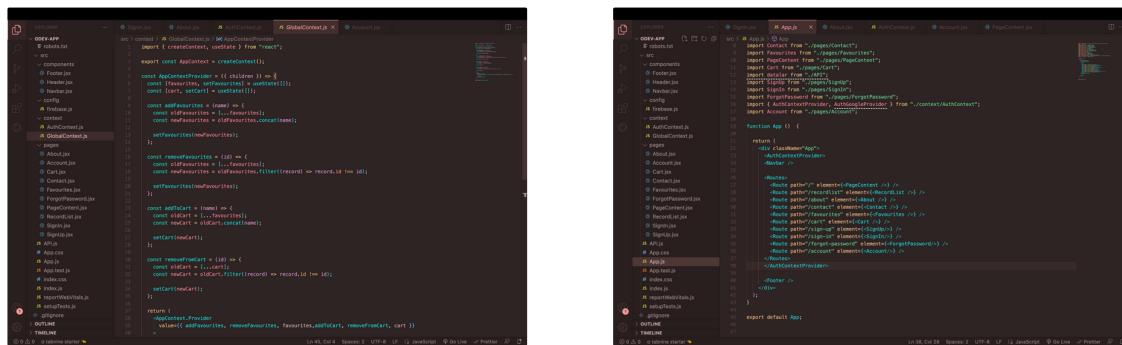
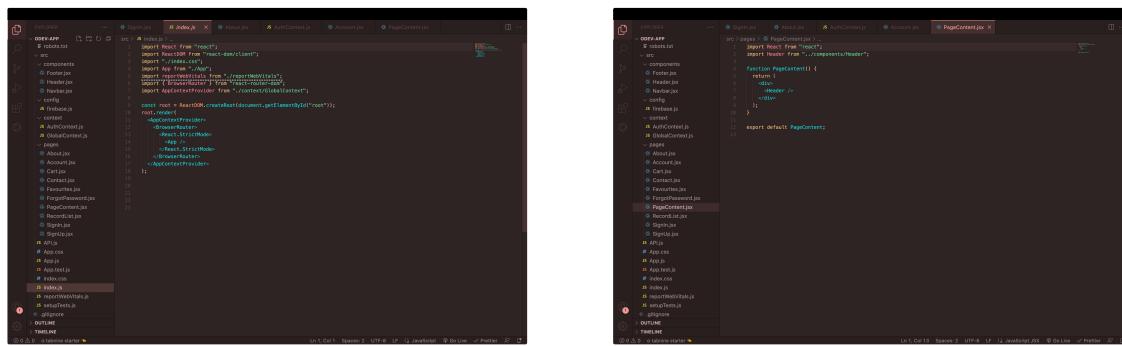


RecordStore

Plak Satış Sitesi

Özet:

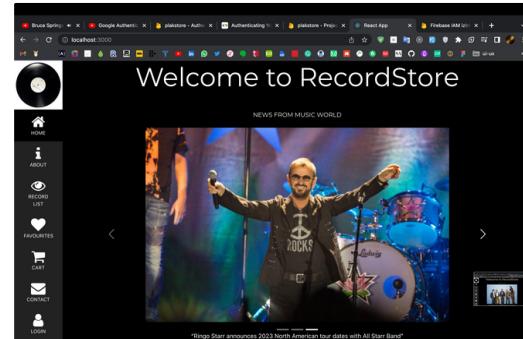
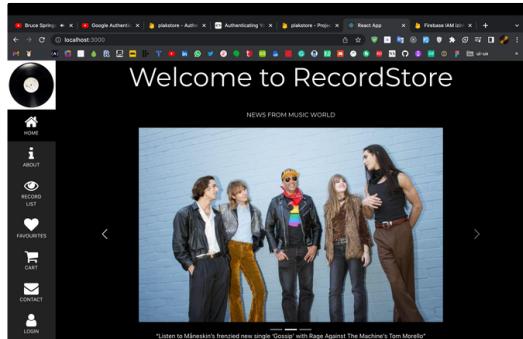
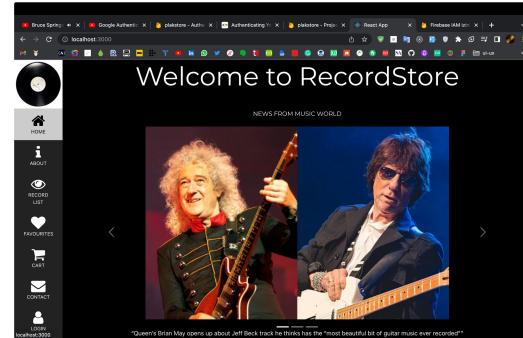
Başlangıçta, src klasörü altında components ve pages isminde iki ayrı klasör oluşturdum. Bu klasörlerin components kısmında Footer, Header, Navbar gibi dosyalarımı toparladım. Pages kısmında ise Favourites, Cart, SignIn ve SignUp gibi sayfalarımı geliştirdim. Api'ı kendim oluştururdum. React ile projemi geliştirdiğim için npm install ile node modulleri yükleyip, Router işlemlerini App.js dosyası altında gerçekleştirdim. Sign-in ve sign-up işlemleri için firebase eklentisi de kurup, config isminde bir klasör oluşturdum. Bu klasör içinde firebase.js isminde bir dosya açıp firebase bağlantısı için gerekli olan config bilgileri ile sign-in ve sign-up işlemleri için fonksiyonları tanımladım. Css için tailwind, w3 ve bootstrap kullandım.



1-Anasayfa

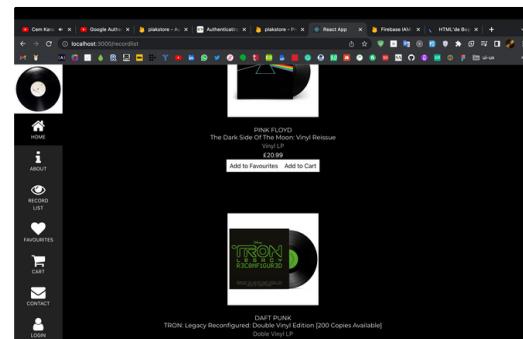
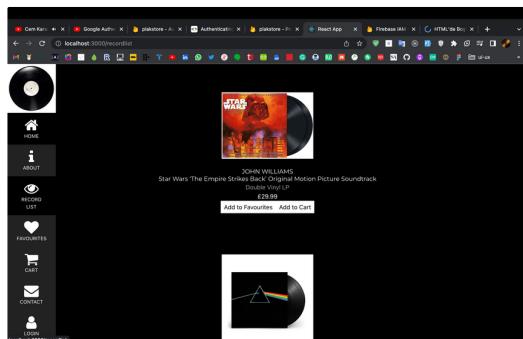
Giriş ekranı aşağıdaki görsellerdeki gibidir. Sayfanın sol kısmındaki navbar sabit olup, her sayfada gelmektedir. Header sayfamda slider kullanarak müzik dünyasından haberler ekledim. Slider hem otomatik, hem de yandaki oklarla hareket ettirilebilmektedir.

```
Compiling successfully!
You can now view dev-app in the browser.
  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.30:3000
Note: the development build is not optimized.
  To create a production build, use:
    yarn build
```



2-Record List

Plakların bulunduğu sayfa burasıdır. Sayfanın genel tasarımını burada da devam ettiğim, Favorilere ekleme ve sepete ekleme olarak iki const oluşturduğum. Return kısmında mapleme yöntemi ile Api üzerindeki plak bilgilerini bu kısımda tanımladım.



```
Compiling successfully!
You can now view dev-app in the browser.
  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.30:3000
Note: the development build is not optimized.
  To create a production build, use:
    yarn build
```

```
function RecordList() {
  const [records, setRecords] = useState([]);
  const [favorites, setFavorites] = useState([]);

  const controlFavorites = (x) => {
    const choice = favorites.some(a => a.id === x);
    return choice;
  }

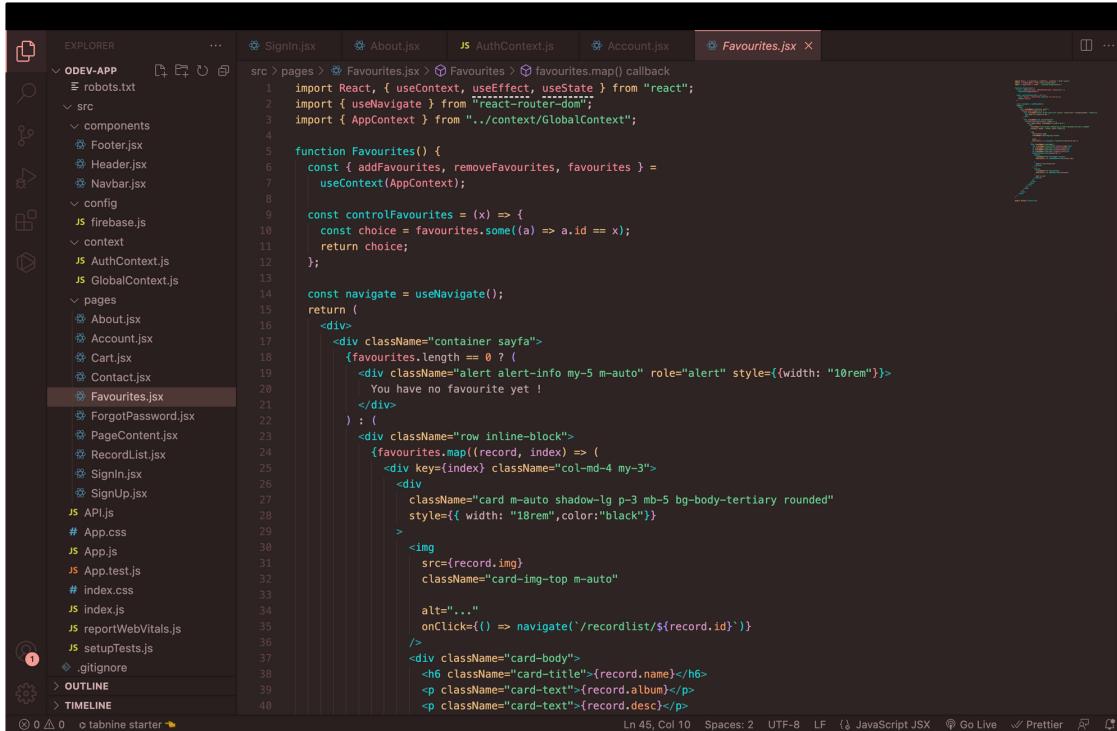
  const addCart = (record) => {
    const choice = cart.some(a => a.id === record.id);
    if (!choice) {
      const cartItem = { ...record, count: 1 };
      setCart([...cart, cartItem]);
    }
  }

  const navigate = useNavigate();
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div><img alt="Record icon" /></div>
              <div>JOHN WILLIAMS<br/>Star Wars: The Empire Strikes Back Original Motion Picture Soundtrack<br/>Double Vinyl LP<br/>£29.99<br/><a href="#">Add to Favorites</a> <a href="#">Add to Cart</a></div>
            </div>
            <div>
              <div><img alt="Record icon" /></div>
              <div>DAFT PUNK<br/>TRON: Legacy Reconfigured Double Vinyl Edition [200 Copies Available]<br/>Double Vinyl LP<br/>£29.99<br/><a href="#">Add to Favorites</a> <a href="#">Add to Cart</a></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
}

export default RecordList;
```

3-Favourites

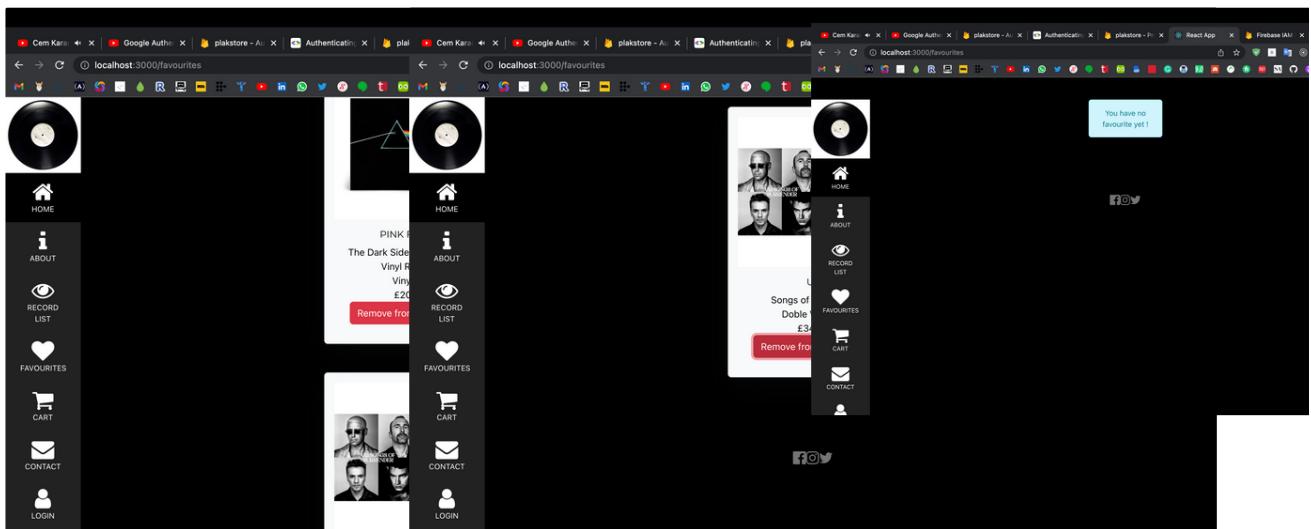
Recordlist sayfasındaki ürünlerden AddtoFav butonuna tıklandığında Favoriler sayfasına eklenmesini sağlamak için usenavigate ve usecontext hook'larını kullandım. Burada tasarım olarak bir card tanımlayıp, RemoveFromFav butonu ekledim. Hiç favori olmadığından bilgi vermesi için bir alert tanımladım.



```

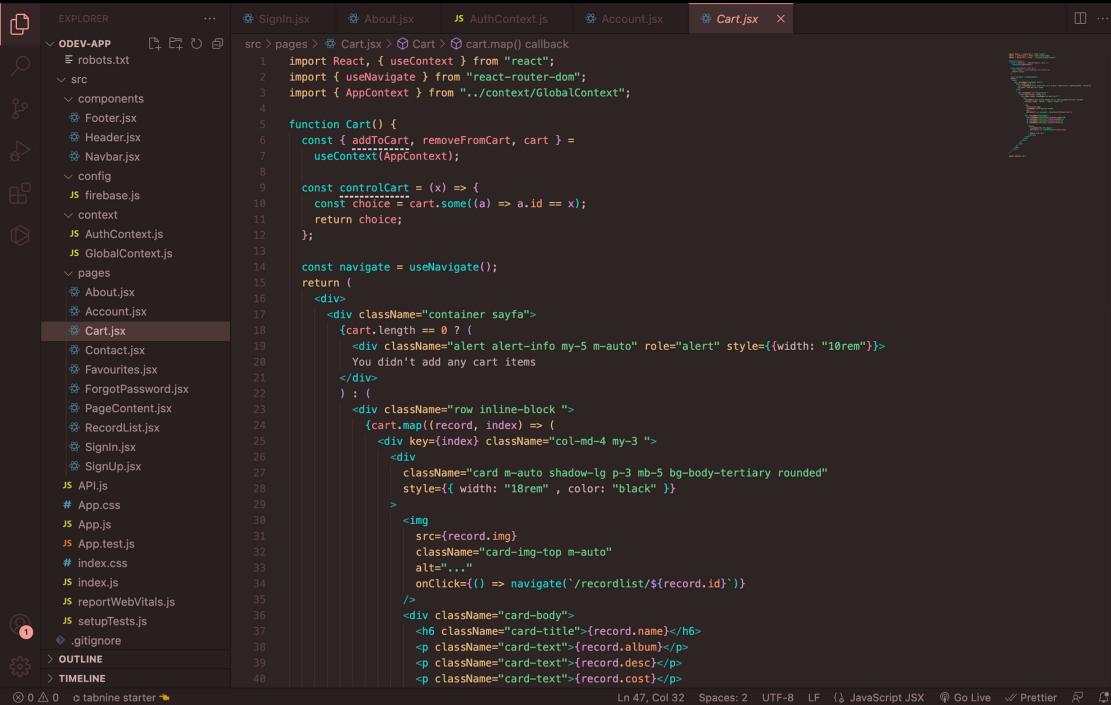
src > pages > Favourites.jsx > Favouredata > favouredata.map().callback
1 import React, { useContext, useEffect, useState } from "react";
2 import { useNavigate } from "react-router-dom";
3 import { AppContext } from "../context/GlobalContext";
4
5 function Favouredata() {
6   const { addFavouredata, removeFavouredata, favouredata } =
7     useContext(AppContext);
8
9   const controlFavouredata = (x) => {
10     const choice = favouredata.some((a) => a.id == x);
11     return choice;
12   };
13
14   const navigate = useNavigate();
15   return (
16     <div>
17       <div className="container sayfa">
18         {favouredata.length == 0 ? (
19           <div className="alert alert-info my-5 m-auto" role="alert" style={{width: "10rem"}}>
20             You have no favourite yet !
21           </div>
22         ) : (
23           <div className="row inline-block">
24             {favouredata.map((record, index) => (
25               <div key={index} className="col-md-4 my-3">
26                 <div className="card m-auto shadow-lg p-3 mb-5 bg-body-tertiary rounded"
27                   style={{width: "18rem", color: "black"}}
28                 >
29                   <img
30                     alt="..." alt={record.img}
31                     onClick={() => navigate(` /recordlist/${record.id}`)}
32                   >
33                   <div className="card-body">
34                     <h6 className="card-title">{record.name}</h6>
35                     <p>{record.album}</p>
36                     <p>{record.desc}</p>
37                   </div>
38               </div>
39             ))}
40           </div>
41         )
42       </div>
43     </div>
44   );
45 }

```



4-Cart

Sepet sayfasında da favoriler sayfasındaki ile benzer constları tanımlayıp, aynı react hook'larını kullandım. (Usenavigate&Usecontext) Tasarım olarak devamlılık olması açısından aynı card'ı kullandım.

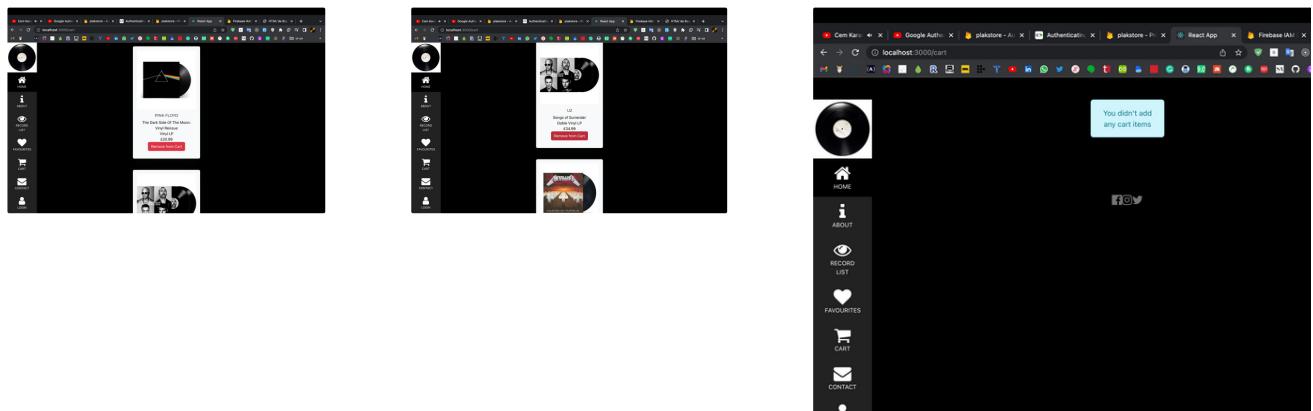


```

src > pages > Cart.jsx > Cart > cart.map() callback
1 import React, { useContext } from "react";
2 import { useNavigate } from "react-router-dom";
3 import { AppContext } from "../context/GlobalContext";
4
5 function Cart() {
6   const [addToCart, removeFromCart, cart] = useContext(AppContext);
7
8   const controlCart = (x) => {
9     const choice = cart.some((a) => a.id === x);
10    return choice;
11  };
12
13
14  const navigate = useNavigate();
15  return (
16    <div>
17      <div className="container sayfa">
18        {cart.length == 0 ? (
19          <div className="alert alert-info my-5 m-auto" role="alert" style={{width: "10rem"}}>
20            You didn't add any cart items
21          </div>
22        ) : (
23          <div className="row inline-block">
24            {cart.map((record, index) => (
25              <div key={index} className="col-md-4 my-3">
26                <div
27                  className="card m-auto shadow-lg p-3 mb-5 bg-body-tertiary rounded"
28                  style={{width: "18rem", color: "black"}}
29                >
30                  <img
31                    src={record.img}
32                    className="card-img-top m-auto"
33                    alt="..."
34                    onClick={() => navigate(`/recordlist/${record.id}`)}
35                  />
36                  <div className="card-body">
37                    <h6>{record.name}</h6>
38                    <p>{record.album}</p>
39                    <p>{record.desc}</p>
40                    <p>{record.cost}</p>
41                </div>
42              </div>
43            ))}
44          </div>
45        )
46      </div>
47    </div>
48  );
49}

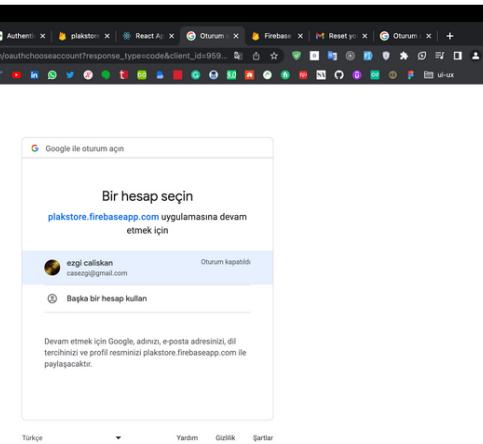
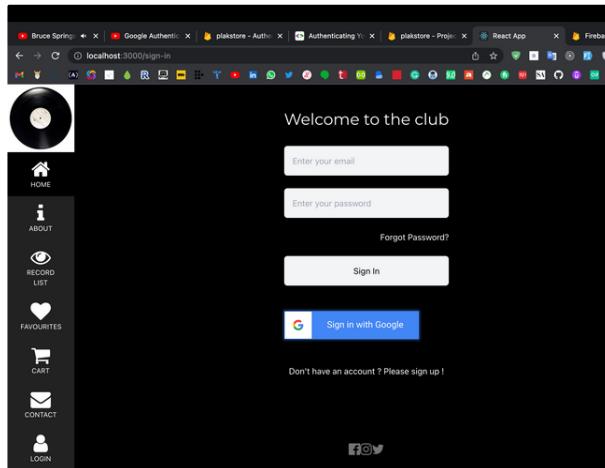
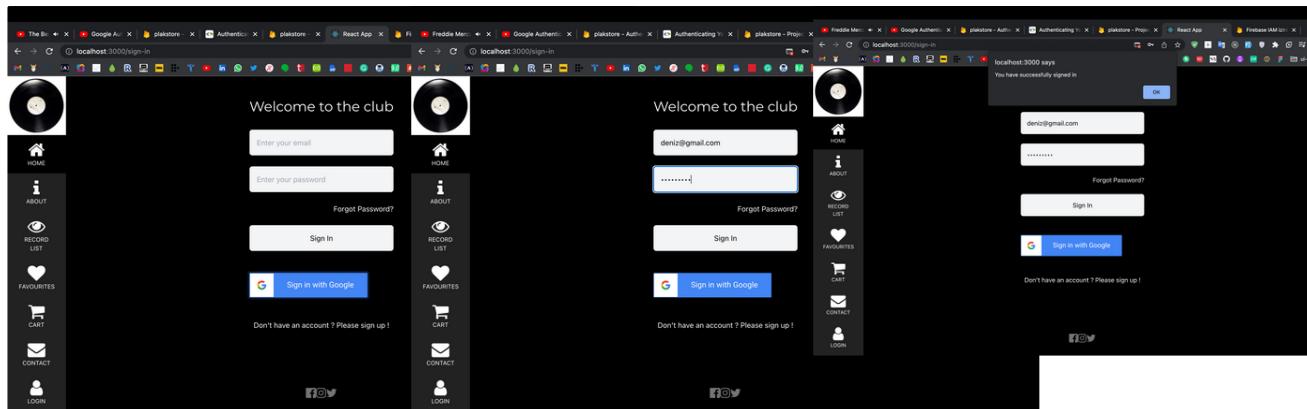
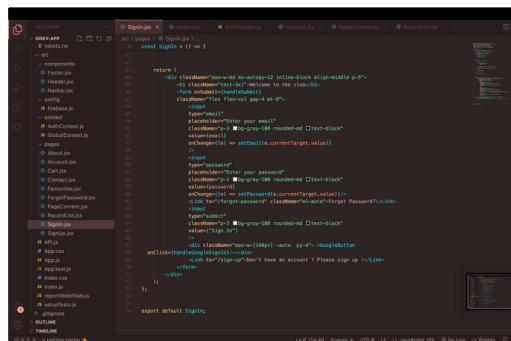
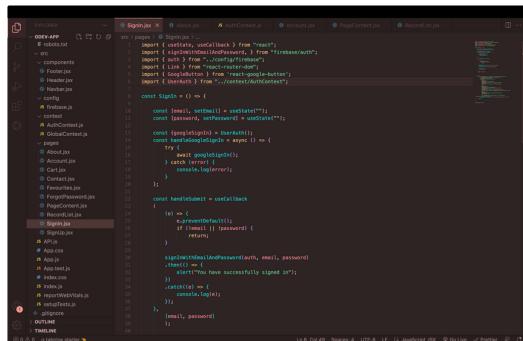
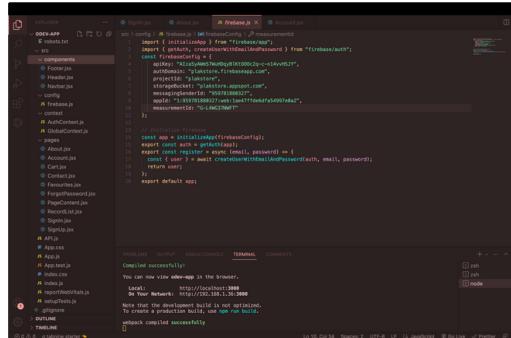
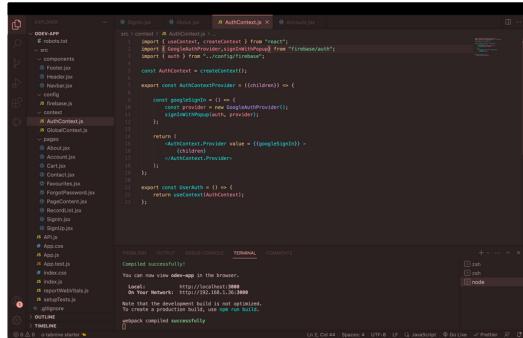
```

Ln 47, Col 32 Spaces: 2 UTF-8 LF ⌂ JavaScript JSX ⌂ Go Live ⌂ Prettier ⌂ ⌂



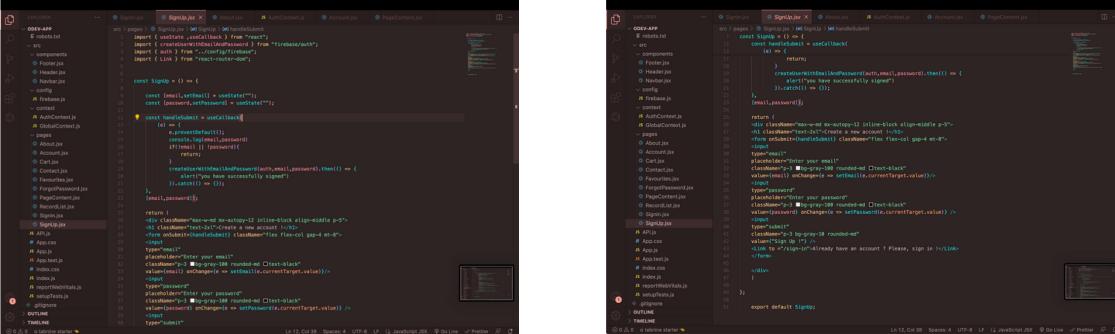
5-Login / Sign-in

E-ticaret sitelerinde üye olmadan da ürünleri görme ve satın alma işlemleri gerçekleşebildiği için, sign-in işlemlerini sayfa açılır açılmaz değil, kişi isteğine bağlı olarak Login kısmında tuttum. Email ve şifre dışında bir de Google ile giriş sağlanması için Context klasörü altında AuthContext isminde bir js dosyası oluşturup, gerekli google authentication komutlarını çağirdım.



6-Sign-Up

Yeni kullanıcı kaydı oluşturulması için mail ve şifre üzere iki parametre alıp firebase/auth üzerinden "createUserWithEmailAndPassword" fonksiyonunu import edip, signup fonksiyonu async hale getirip bu fonksiyon içinde çağrırdım ve gerekli parametrelerini tanımladım. Yeni kullanıcı yaratılması başarılı olduktan sonra alert belirip, bunu kullanıcıya bildirdim. Sonrasında Firebase üzerinden girilen mail ve şifre bilgilerini kontrol ettim.



The image shows two side-by-side code editors. The left editor displays a file named `Signin.js` containing React component code. The right editor displays a file named `Signup.js` containing Firebase authentication logic. Both files are part of a React application structure with components like `Header.js`, `Footer.js`, and `AuthContext.js`.

```

// Signin.js
import React from 'react';
import { useState } from 'react';
import { createUserWithEmailAndPassword } from 'firebase/auth';
import { Link } from 'react-router-dom';

const Signin = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleEmailChange = (e) => {
    setEmail(e.target.value);
  };

  const handlePasswordChange = (e) => {
    setPassword(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const user = createUserWithEmailAndPassword(auth, email, password);
    user.then(() => {
      alert('You have successfully signed up');
    }).catch((err) => {
      console.error(err.message);
    });
  };
}

export default Signin;

// Signup.js
import React from 'react';
import { useState } from 'react';
import { createUserWithEmailAndPassword } from 'firebase/auth';
import { Link } from 'react-router-dom';

const Signup = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleEmailChange = (e) => {
    setEmail(e.target.value);
  };

  const handlePasswordChange = (e) => {
    setPassword(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const user = createUserWithEmailAndPassword(auth, email, password);
    user.then(() => {
      alert('You have successfully signed up');
    }).catch((err) => {
      console.error(err.message);
    });
  };
}

export default Signup;

```

```

// Signup.js
import React from 'react';
import { useState } from 'react';
import { createUserWithEmailAndPassword } from 'firebase/auth';
import { Link } from 'react-router-dom';

const Signup = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

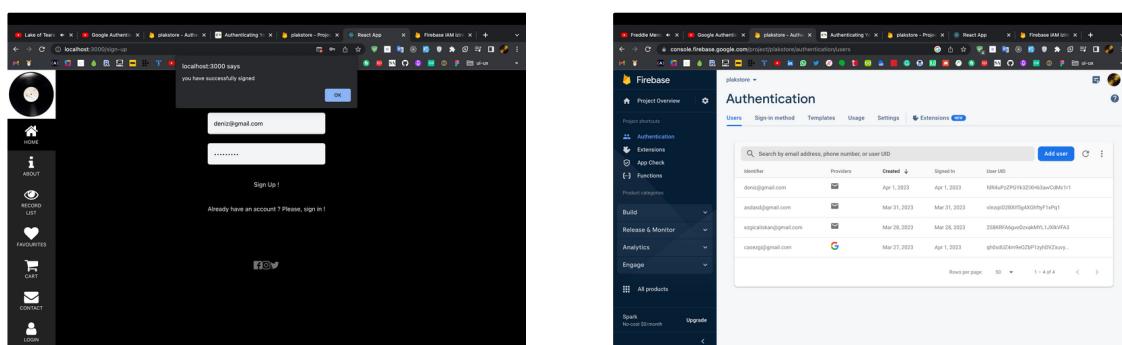
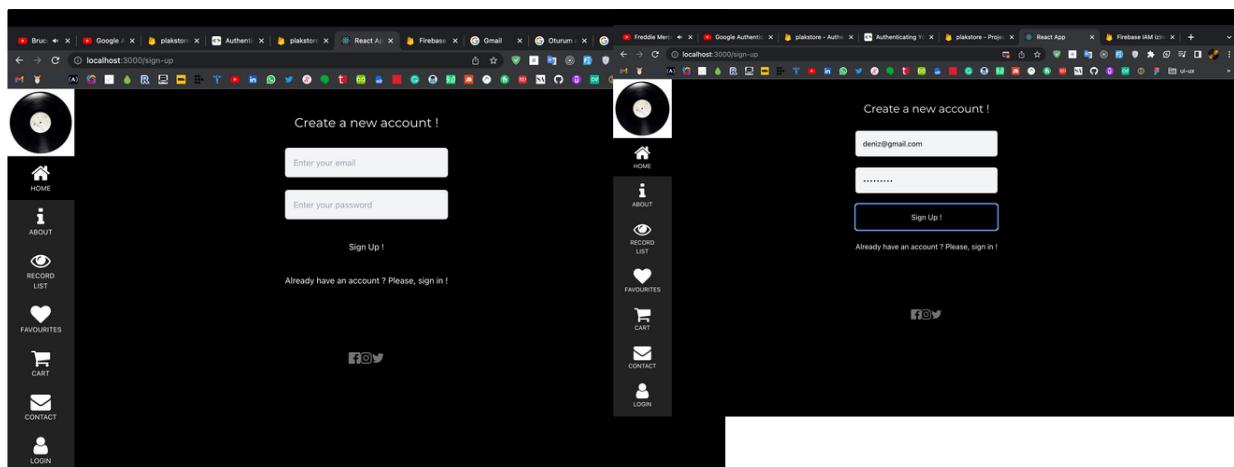
  const handleEmailChange = (e) => {
    setEmail(e.target.value);
  };

  const handlePasswordChange = (e) => {
    setPassword(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const user = createUserWithEmailAndPassword(auth, email, password);
    user.then(() => {
      alert('You have successfully signed up');
    }).catch((err) => {
      console.error(err.message);
    });
  };
}

export default Signup;

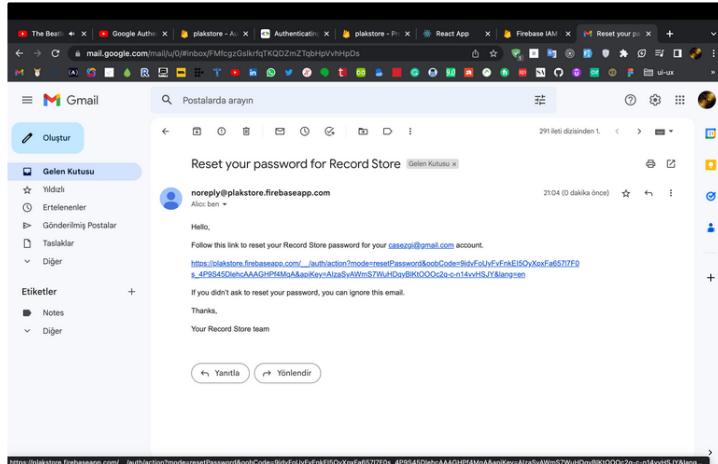
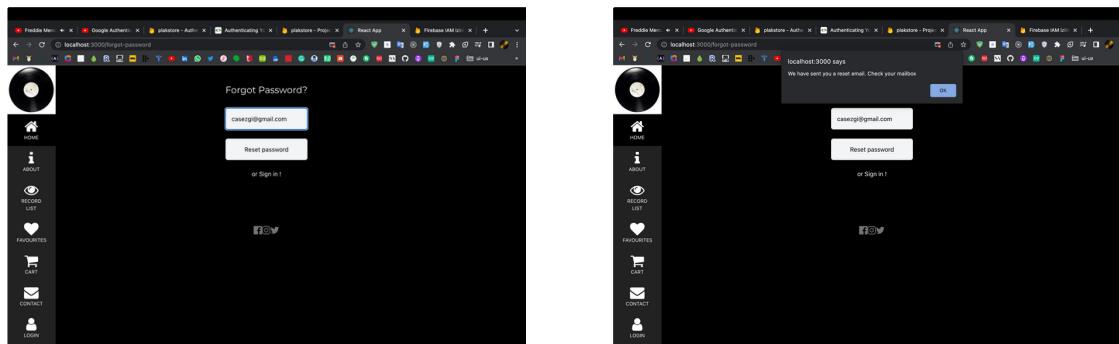
```



7-Forgot Password

Sign-in sayfasına şifremi unuttum seçeneği ekleyip girilen maile link yollama işlemini de yine firebase ile gerçekleştirdim.

```
src/pages/ForgotPassword.jsx
1 import { useState } from 'react';
2 import { Form, Input, Button } from 'antd';
3 import { sendPasswordResetEmail } from 'firebase/auth';
4 import { auth } from '../config/firebase';
5
6 const ForgetPassword = () => {
7
8     const [email, setEmail] = useState('');
9
10    const handleSubmit = usecallback((e) => {
11        e.preventDefault();
12
13        if (email) {
14            return;
15        }
16
17        sendPasswordResetEmail(auth, email).then(() => {
18            alert('We have sent you a reset email. Check your mailbox!');
19        }).catch((e) => {
20            console.log(e);
21        });
22    }, [email]);
23
24    return (
25        <div className="max-w-md mx-auto py-10 flex flex-col align-middle">
26            <h3>Forgot Password</h3>
27            <form onSubmit={handleSubmit} className="flex flex-col gap-4 mt-8">
28                <input type="email" placeholder="Enter your email" className="p-3 bg-gray-100 rounded-md outline-none border-0" />
29                <input type="submit" value="Reset password" className="p-3 bg-gray-100 rounded-md outline-none border-0" />
30                <a href="#" type="button">Or Sign In</a>
31            </form>
32        </div>
33    );
34
35}
36
37 export default ForgetPassword;
```



8-Footer

Son olarak footer üzerindeki sosyal medya yönlendirmelerini ekledim.

