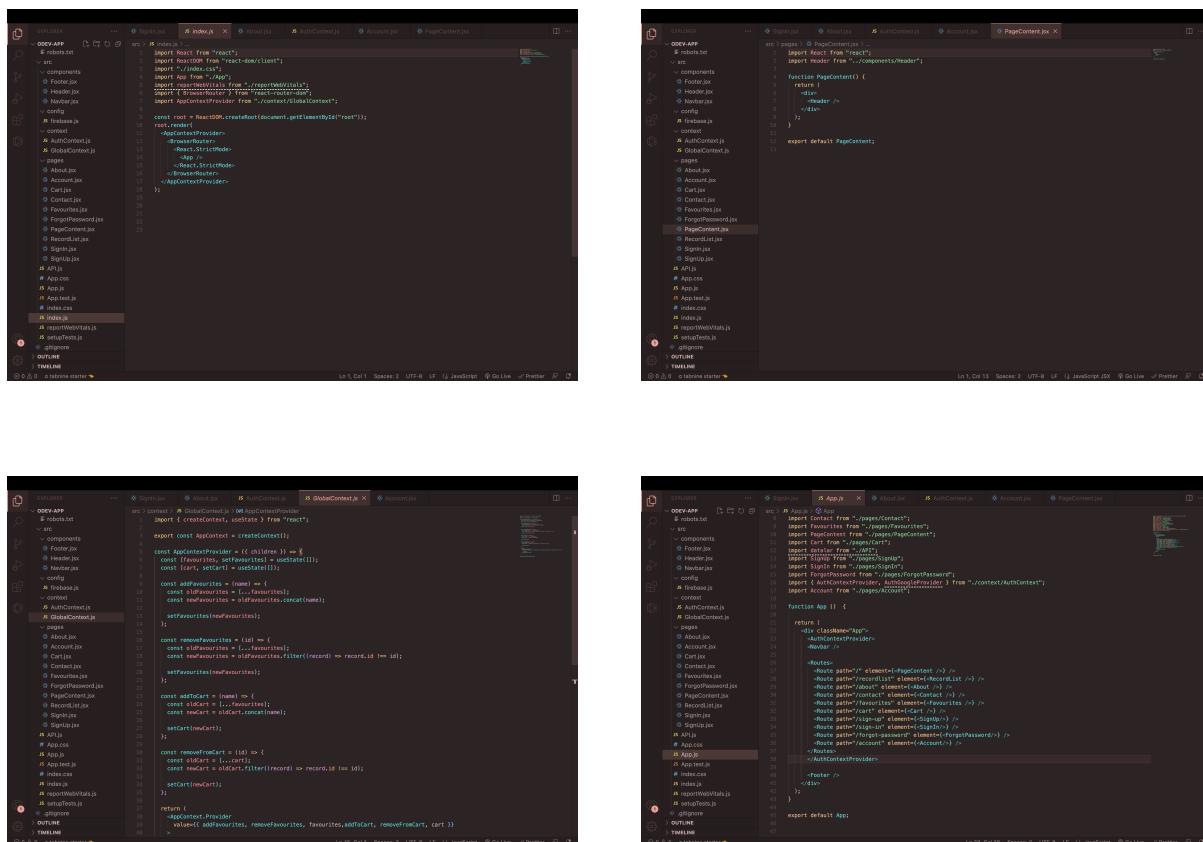


RecordStore

Plak Satış Sitesi

Özet:

Başlangıçta, src klasörü altında components ve pages isminde iki ayrı klasör oluştururdum. 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 oluştururdum. 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

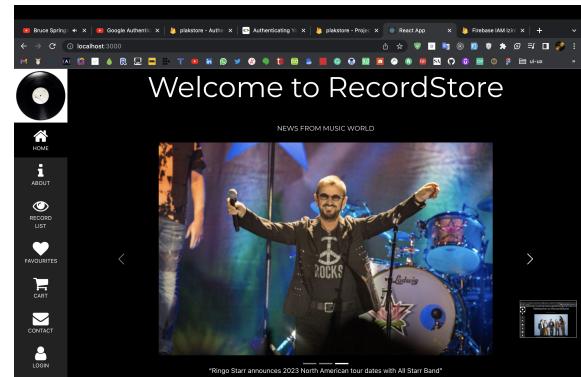
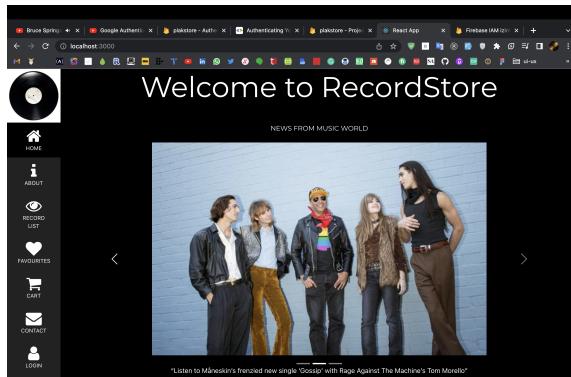
dünyasından haberler ekledim. Slider hem otomatik, hem de yandaki oklarla hareket ettirilebilmektedir.

```

import React from 'react';

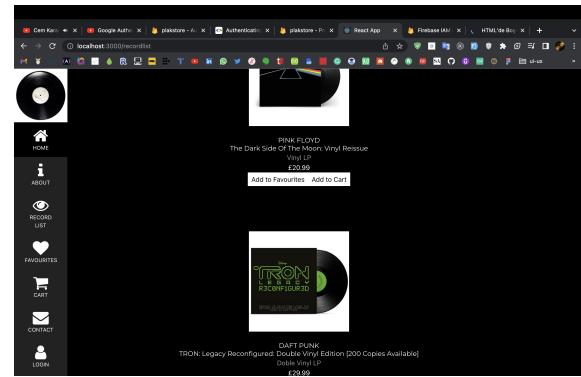
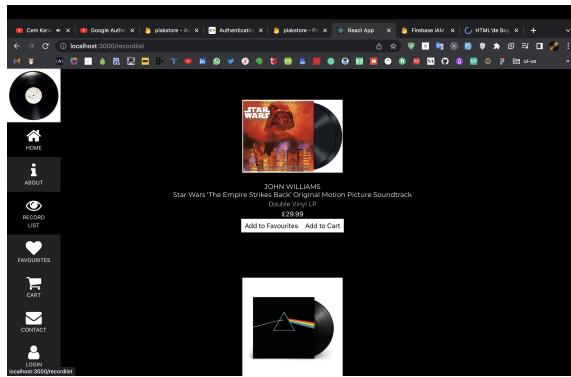
function Header() {
  return (
    <div>
      <div>
        <div>
          <div><img alt="Logo" /></div>
          <div>HOME</div>
          <div>ABOUT</div>
          <div>RECORD LIST</div>
          <div>FAVOURITES</div>
          <div>CONTACT</div>
        </div>
        <div>
          <input type="text" placeholder="Search..." />
        </div>
      </div>
    </div>
  );
}

export default Header;
  
```



2-Record List

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





```
src/api/Records.js
import React, { useState } from 'react';
import { RecordList, RecordItem } from './RecordList';
import { RecordForm } from './RecordForm';

const RecordsList = () => {
  const [records, setRecords] = useState([]);
  const [selectedRecord, setSelectedRecord] = useState(null);
  const [isEditing, setIsEditing] = useState(false);

  const handleAddRecord = (record) => {
    setRecords([record, ...records]);
  };

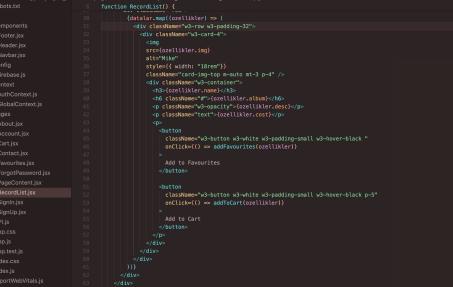
  const handleEditRecord = (record) => {
    setSelectedRecord(record);
    setIsEditing(true);
  };

  const handleDeleteRecord = (id) => {
    const newRecords = records.filter(r => r.id !== id);
    setRecords(newRecords);
  };

  const handleSaveRecord = (record) => {
    const updatedRecords = records.map(r => r.id === record.id ? record : r);
    setRecords(updatedRecords);
  };

  return (
    <div>
      <h2>Records List</h2>
      <table border="1">
        <thead>
          <tr>
            <th>Name</th>
            <th>Category</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          {records.map((r) => (
            <tr key={r.id}>
              <td>{r.name}</td>
              <td>{r.category}</td>
              <td>
                <button onClick={()=>handleEditRecord(r)}>Edit</button>
                <button onClick={()=>handleDeleteRecord(r)}>Delete</button>
              </td>
            </tr>
          ))}
        </tbody>
      </table>
      {selectedRecord ? (
        <RecordForm record={selectedRecord} onEdit={handleSaveRecord} onReset={handleEditRecord} onDelete={handleDeleteRecord} />
      ) : null}
    </div>
  );
};

export default RecordsList;
```



```
src/components/RecordList.js
import React, { useState } from 'react';
import { RecordContext } from './RecordContext';
import RecordItem from './RecordItem';

function RecordList() {
  const [selectedRecord, setSelectedRecord] = useState(null);
  const handleSelectRecord = (record) => {
    setSelectedRecord(record);
  };
  const handleDeleteRecord = (record) => {
    // Logic to delete record
  };

  return (
    <div>
      <h2>Record List</h2>
      <p>Selected Record: {selectedRecord ? selectedRecord.name : null}</p>
      <ul>
        {records.map((record) => (
          <RecordItem
            key={record.id}
            name={record.name}
            onClick={handleSelectRecord}
            onDelete={handleDeleteRecord}
          />
        ))}
      </ul>
    </div>
  );
}

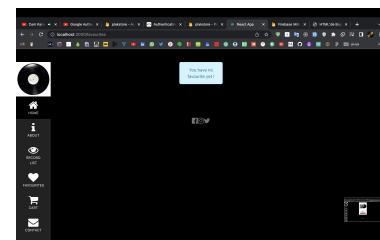
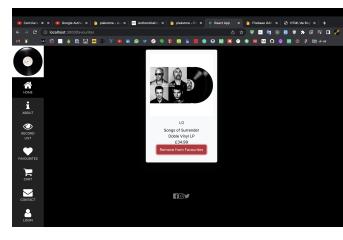
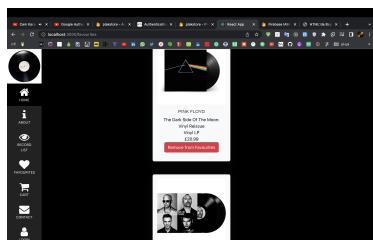
const records = [
  { id: 1, name: 'Record A' },
  { id: 2, name: 'Record B' },
  { id: 3, name: 'Record C' },
  { id: 4, name: 'Record D' },
  { id: 5, name: 'Record E' },
];

export default RecordList;
```

3-Favourites

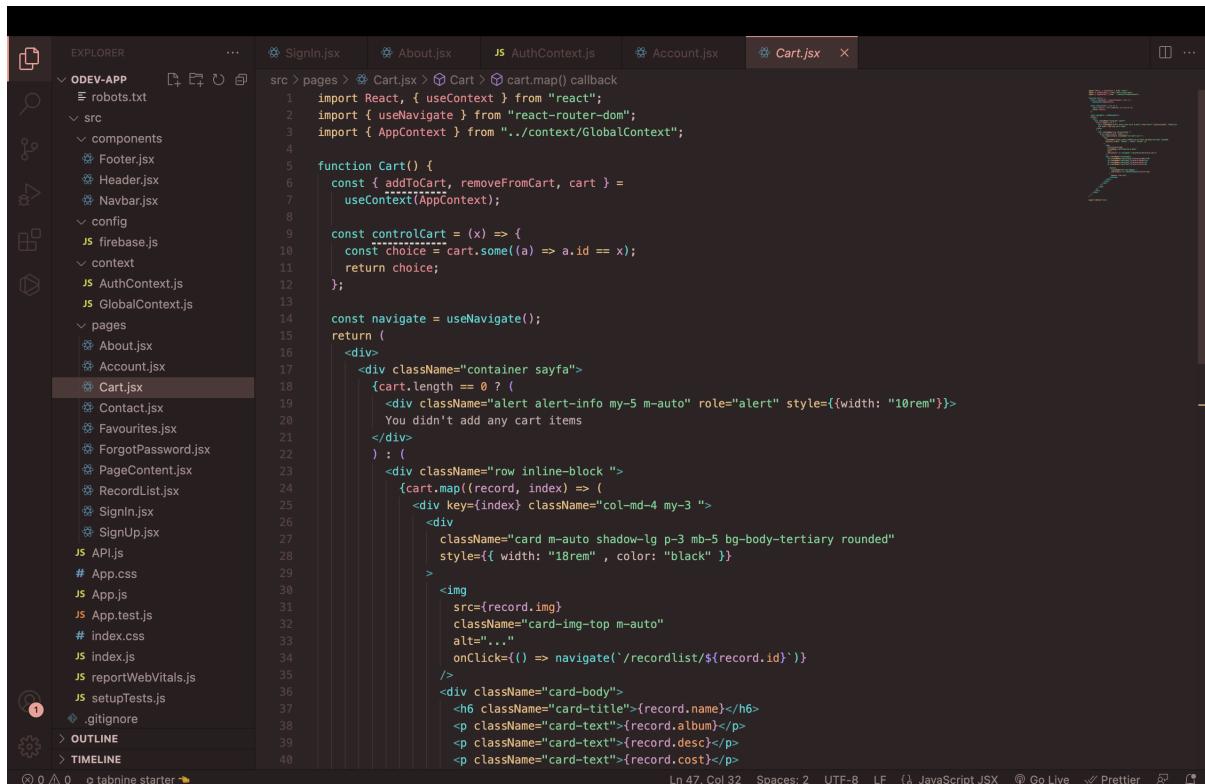
Recordlist sayfasındaki ürünlerden AddToFav butonuna tıklandığında Favoriler sayfasına eklenmesini sağlamak için usenavigate ve usecontext hook'larını kullandım. Burada tasarım olarak bir card tanımlayıp, RemoveFromFav butonu ekledim. Hiç favori olmadığında bilgi vermesi için bir alert tanımladım.

```
src > pages > Favourites.jsx > Favourites > favourites.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 Favourites() {
  6   const { addFavourites, removeFavourites, favourites } =
  7     useContext(AppContext);
  8
  9   const controlFavourites = (x) => {
 10     const choice = favourites.some((a) => a.id === x);
 11     return choice;
 12   };
 13
 14   const navigate = useNavigate();
 15   return (
 16     <div>
 17       <div className="container sayfa">
 18         {favourites.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             {favourites.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
 34                     alt="..."
 35                     onClick={() => navigate(`'/recordlist/${record.id}`)}
 36                   />
 37                   <div className="card-body">
 38                     <h6 className="card-title">{record.name}</h6>
 39                     <p className="card-text">{record.album}</p>
 40                     <p className="card-text">{record.desc}</p>
 41               </div>
 42             </div>
 43           </div>
 44         )}
 45       </div>
 46     </div>
 47   );
 48 }
 49
 50 export default Favourites;
```

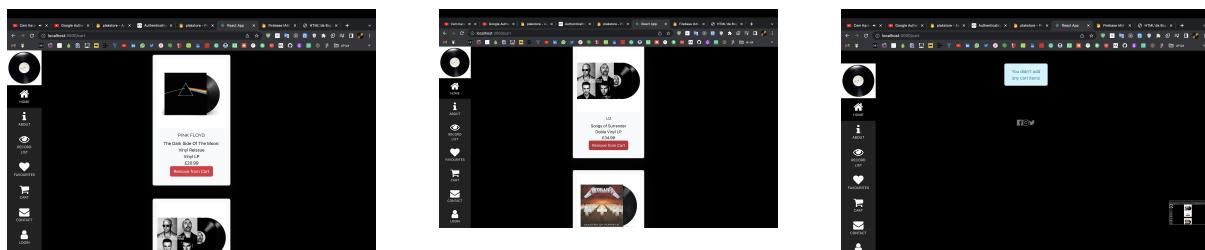


4-Cart

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 } =
7     useContext(AppContext);
8
9   const controlCart = (x) => {
10     const choice = cart.some((a) => a.id === x);
11     return choice;
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 className="card m-auto shadow-lg p-3 mb-5 bg-body-tertiary rounded"
27                   style={{ width: "18rem", color: "black" }}>
28                   <img
29                     src={record.img}
30                     className="card-img-top m-auto"
31                     alt="..."
32                     onClick={() => navigate(`/${record.id}`)}>
33                   </div>
34                   <div className="card-body">
35                     <h6 className="card-title">{record.name}</h6>
36                     <p className="card-text">{record.album}</p>
37                     <p className="card-text">{record.desc}</p>
38                     <p className="card-text">{record.cost}</p>
39                   </div>
40                 </div>
41               ) >
42             </div>
43           </div>
44         )
45       </div>
46     </div>
47   );
48 }
49
50 <div>
51   <Cart />
52 </div>
```



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 çağrırdım.



The screenshot shows a terminal window with several tabs open. The current tab displays the command `npm start` and its output, which includes the message "Compiled successfully!". It also shows the URL `http://127.0.0.1:3000` and the text "On Your Network". Below this, there is a note about the development build not being optimized. The bottom of the terminal shows the command `webpack compiled successfully`. The background of the terminal shows a React application with a sidebar menu and a main content area.

```
firebase deploy --only hosting

  ✔  Deploy successful!
  Your new Firebase app is available at:
  https://eden-app-12345.firebaseio.com
  On Your Network
  Note that the development build is not optimized.
  This build is intended for deployment to the cloud.
  Run 'npm start' to see a production-optimized build.
  webpack compiled successfully

  + 20s
  + 20s
  + 20s
  + 20s
```

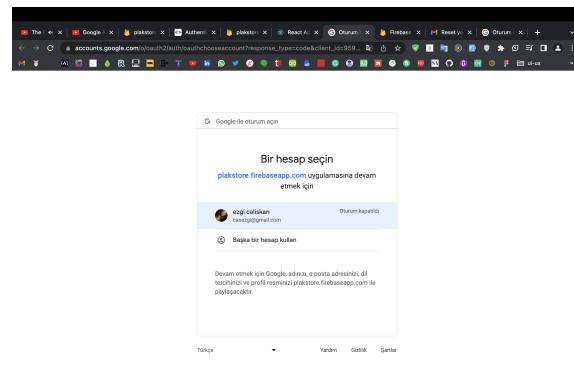
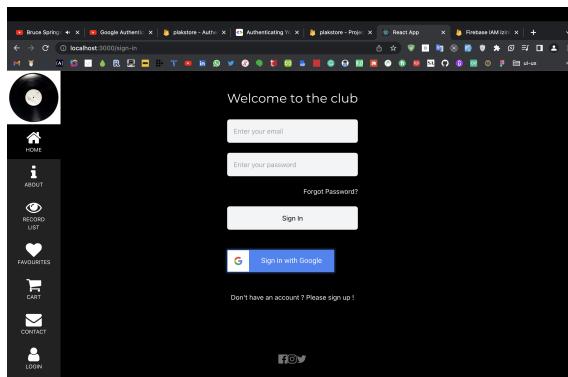
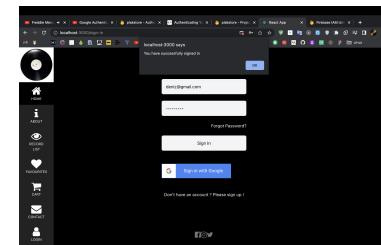
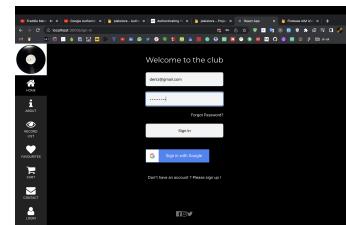
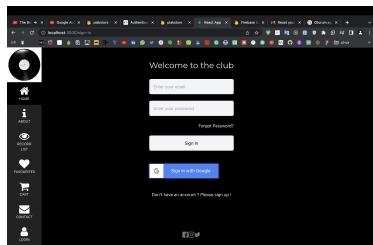


```
import React from 'react';
import { Formik, Form, Field, ErrorMessage } from 'formik';
import * as Yup from 'yup';
import { Link } from 'react-router-dom';
import { Container, Row, Col, FormText, FormFeedback } from 'reactstrap';

const Signin = () => {
  const schema = Yup.object().shape({
    email: Yup.string()
      .email('Email is not valid')
      .required('Email is required'),
    password: Yup.string()
      .min(6, 'Password must be at least 6 characters')
      .required('Password is required'),
  });

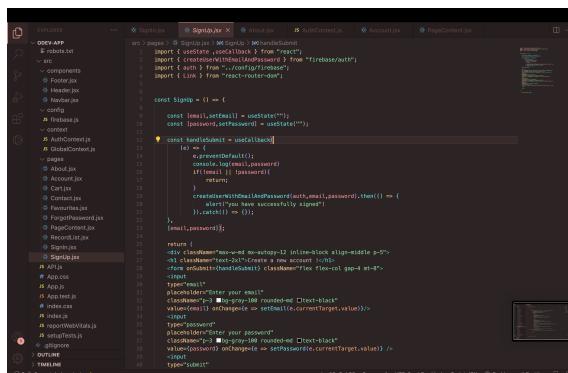
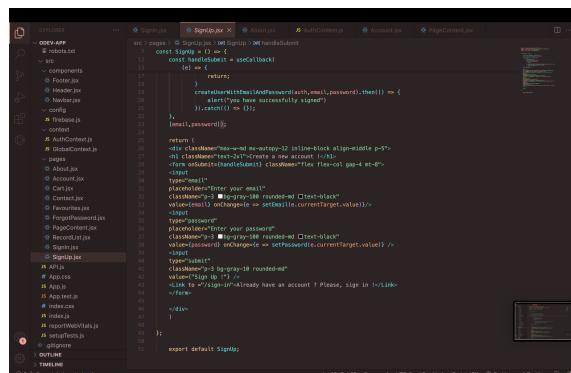
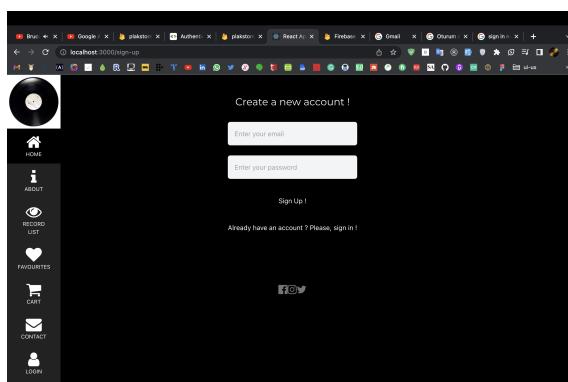
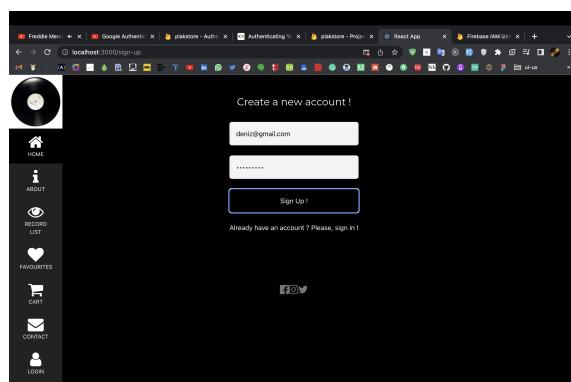
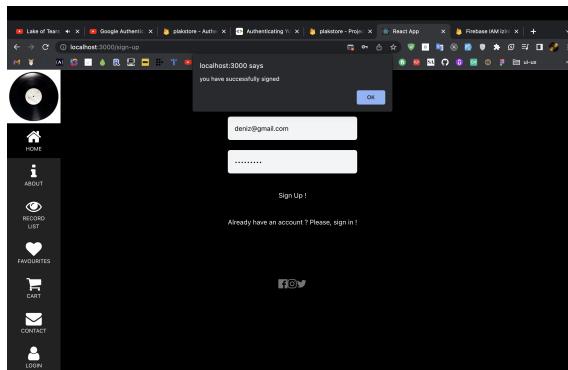
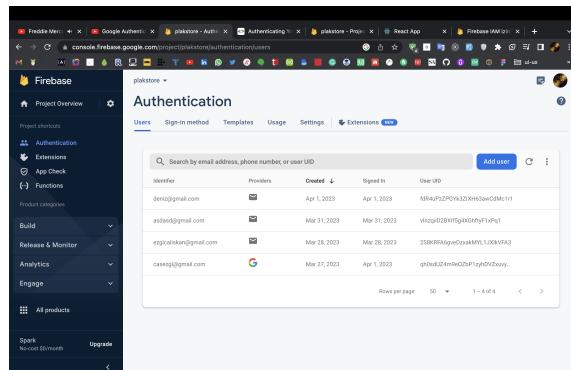
  return (
    <Container fluid="true" style={{ margin: 'auto', width: 'fit-content' }}>
      <Row style={{ align-items: 'center' }}>
        <Col style={{ flex: 1 }}>
          <Formik
            initialValues={{ email: '', password: '' }}
            validationSchema={schema}
            onSubmit={(values) => console.log(values)}
          >
            <Form>
              <div style={{ display: 'flex', align-items: 'center' }}>
                <input type="text" name="email" placeholder="Enter your email!" style={{ border: '1px solid #ccc', padding: '5px' }} />
                <FormText style={{ margin: '0 10px' }}>Forgot Password?</FormText>
              </div>
              <div style={{ display: 'flex', align-items: 'center' }}>
                <input type="password" name="password" placeholder="Enter your password!" style={{ border: '1px solid #ccc', padding: '5px' }} />
                <FormText style={{ margin: '0 10px' }}>Forgot Password?</FormText>
              </div>
              <div style={{ display: 'flex', align-items: 'center' }}>
                <input checked="" type="checkbox" name="rememberMe" style={{ border: '1px solid #ccc', padding: '5px' }} />
                <FormText style={{ margin: '0 10px' }}>Remember Me</FormText>
                <Link href="#" style={{ color: '#007bff', textDecoration: 'underline' }}>Forgot Password?</Link>
              </div>
              <div style={{ display: 'flex', align-items: 'center' }}>
                <input type="button" value="Sign In" style={{ border: '1px solid #007bff', padding: '5px 10px', background: '#007bff', color: 'white', cursor: 'pointer' }} />
                <FormText style={{ margin: '0 10px' }}>Don't have an account? Please sign up <Link href="#" style={{ color: '#007bff', textDecoration: 'underline' }}>Here!</Link></FormText>
              </div>
            </Form>
          </Formik>
        </Col>
      </Row>
    </Container>
  );
}

export default Signin;
```



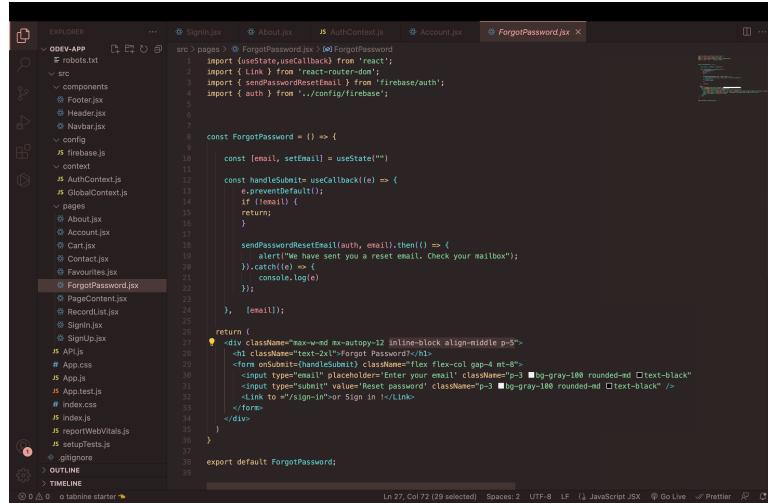
6-Sign-Up

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.

7-Forgot Password

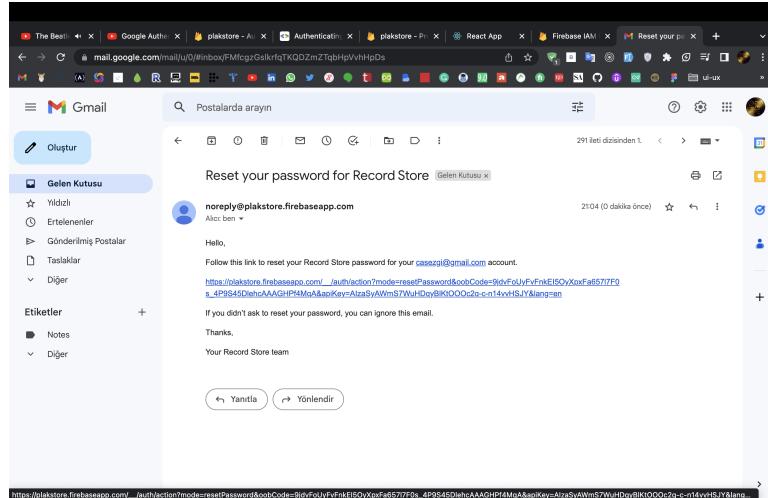
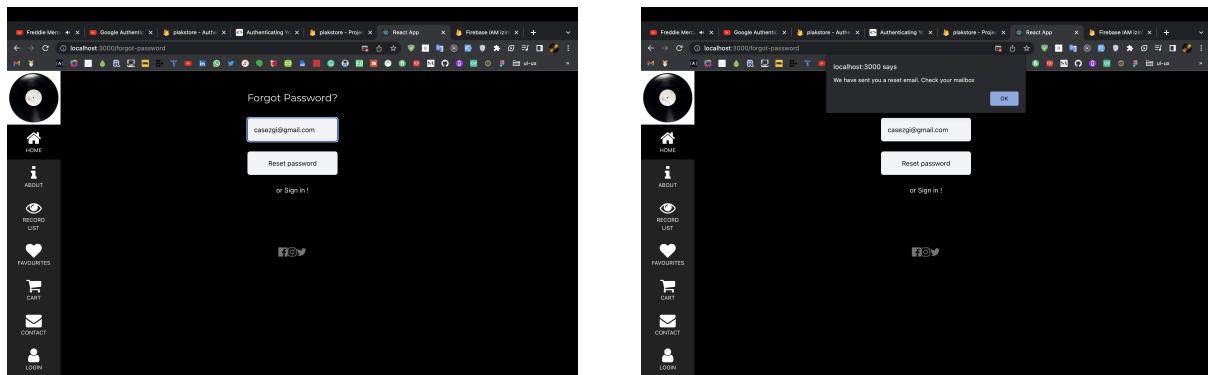
Sign-in sayfasına şifremi unuttum seçeneği ekleyip girilen maile link yollama işlemini de yine firebase ile gerçekleştirdim.



```

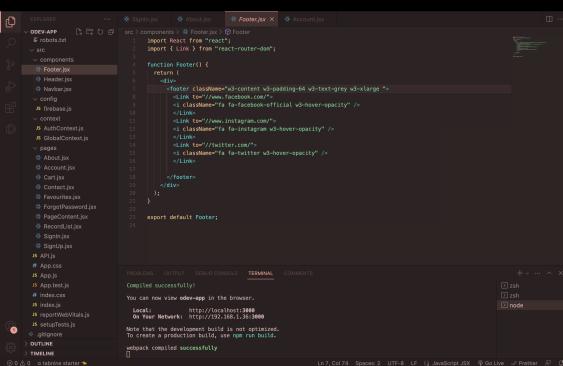
src > pages > ForgotPassword.jsx
  <div>
    <form onSubmit={handleSubmit}>
      <input type="text" placeholder="Enter your email" />
      <button type="submit">Reset Password</button>
    </form>
  </div>
  <div>
    <p>We have sent you a reset email. Check your mailbox!</p>
  </div>

```



8-Footer

Son olarak footer üzerindeki sosyal medya yönlendirmelerini ekledim.



```
import React from 'react';
import { Link } from 'react-router-dom';
function Footer() {
  return (
    <div>
      <a href="#">@recordstoreofficial</a>
      <a href="https://www.facebook.com/">Facebook</a>
      <a href="https://www.twitter.com/">Twitter</a>
      <a href="https://www.instagram.com/">Instagram</a>
    </div>
  );
}
export default Footer;
```

