

WEB PROGRAMLAMA FİNAL ÖDEVİ

2. El Eşya Uygulaması

Uygulama linki: <https://ikinci-el-esya-adae9.web.app/>

Uygulama Github linki: <https://github.com/alperenkapusuz/ikinci-el-esya>

Alperen Kapusuz

1191602061

Bilgisayar Mühendisliği

Uygulama tanımı

- Uygulamamız 2. el eşya alım-satım üzerine kurulmuştur
- Kullanıcı uygulamamıza Google hesabı ile giriş yaparak uygulama içi alım-satım işlemlerine başlayabilir.
 - Giriş yapan kullanıcı anasayfada 2. el satılık eşyaları görebilir
 - Ürünleri kategorileri, konum ve isimlerine göre filtreleyebilir
 - Ürün kartında ürünün adını, fiyatını konumunu görebilir
 - Ürün kartında detay butonuna basarak o ürün hakkında daha detaylı bilgiler edinebilir ve kullanıcıya mesaj atabilir

Uygulama tasarımını kullanıcı dostu ve sonuç odaklı olacak şekilde tasarladım

Anasayfa tasarımı

ikinciel

Ürün Ekle

Çıkış Yap

Google ile giriş yap

Category filter


Place filter

Search


Category

Place


iPhone



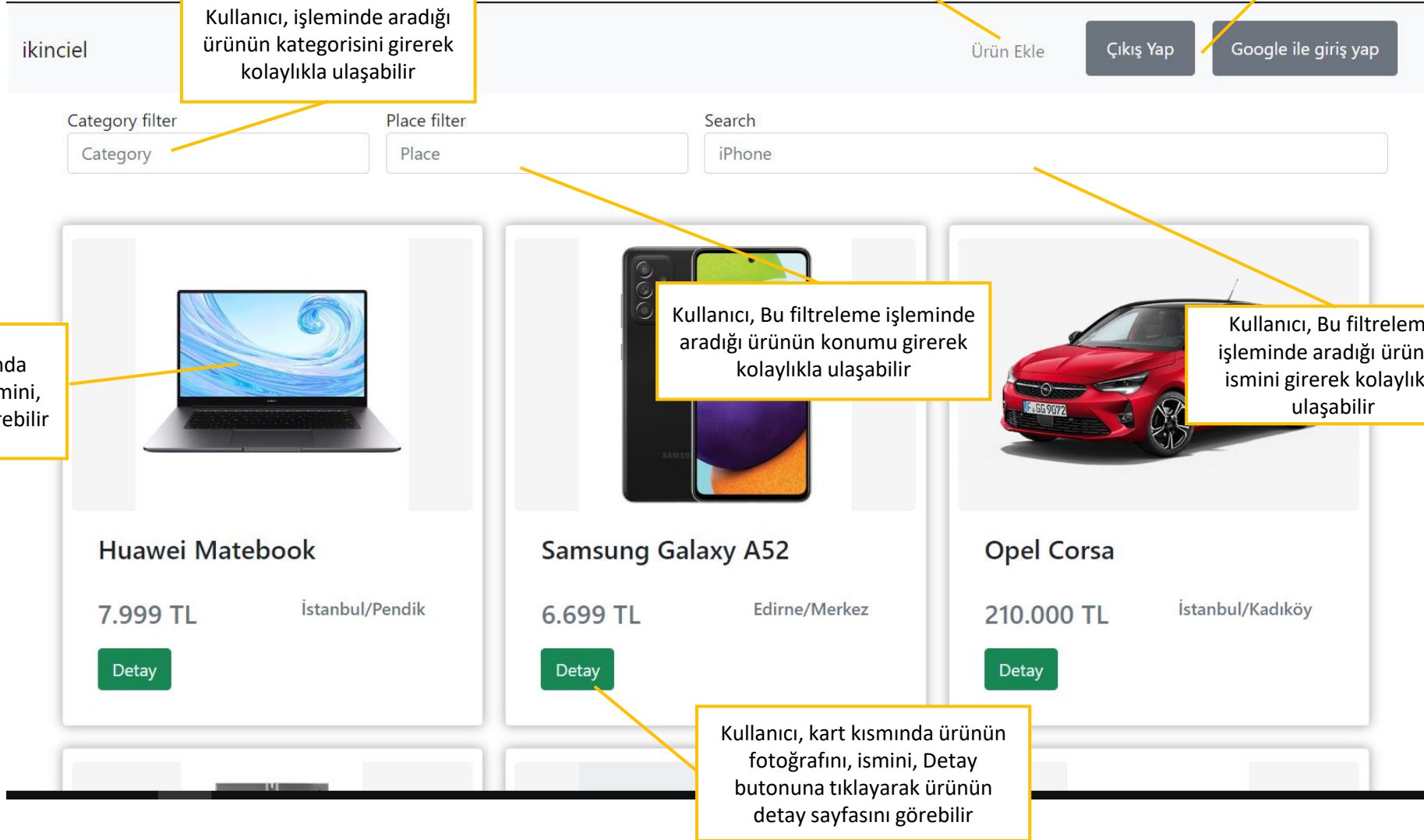
Huawei Matebook
7.999 TLİstanbul/Pendik
[Detay](#)



Samsung Galaxy A52
6.699 TLİzmir/Merkez
[Detay](#)



Opel Corsa
210.000 TLİstanbul/Kadıköy
[Detay](#)



Detay sayfası tasarımı

ikinci

Ürün Ekle

Çıkış Yap

Google ile giriş yap



Cihat Öztürk

cihatozturk@gmail.com

Mesaj Gönder

Futbol Topu

Spor

Tokat/Niksar

Price: ₺1.899

Anasayfaya dön

%70 poliüretan, %20 geri dönüştürülmüş polyester, %10 viskon kaplama


Detay sayfası tasarımı


ikinci

Ürün Ekle

Çıkış Yap

Google ile giriş yap





Cihat Öztürk
cihatozturk@gmail.com

Mesaj Gönder

Futbol Topu

SporTokat/Niksar

Price: ₺1.899

Anasayfaya dön

%70 poliüretan, %20 geri dönüştürülmüş polyester, %10 viskon kaplama

Bu kısım ürünün açıklamasını bulunduğu kısımdır

Resimlerin gösterildiği bu kısımda Carousel kullanıldı, kullanıcı ürünün resimlerini buradan görebilir

Bu kısımda satıcının resmi, Gmail hesabı ve mesaj göndermek için bir buton bulunur

Bu kısımda ürünün ismi, kategorisi, konumu ve fiyatı bulunur

Mesajlaşma bölümü

ikinci1

Ürün Ekle

Çıkış Yap

Google ile giriş yap

Message...

Send

S kaç gayme

son oluru 11.000

merhaba

Ürün Ekleme bölümü

ikinci1

Ürün Ekle

Çıkış Yap

Google ile giriş yap

Title

Enter title

Category

Enter title

Price

Enter price

Place

Enter place

Image 1

Enter image URL

Image 2

Enter image URL

Image 3

Enter image URL

Description

Enter description

EKLE

App.js component

- Bu komponent, uygulamamızın ana sayfasını çalıştıran komponentdir.
- Uygulamamda react-router-dom kullandım. Böylelikle komponentler arası geçişi sağlamış oldum.
- getItem fonksiyonunda, oluşturduğum localStorage'dan dataları aldım. LocalStorage boş durumda ise oluşturduğum dummy data'yı local storage'a push ettim, dolu durumda ise localdata parse ederek objeye dönüştürdüm.
- Return kısmında iste dom işlemlerini uyguladım.
 - Her sayfada oluşması için Navi komponentini Routes dışına koydum.
 - Routes kısmında içerisinde ise gidilecek komponentlerin path kısımlarını koydum.

```
src > App.js > App
1  import React, { useEffect, useState } from 'react'
2  import Navi from '../components/Navbar/Navi'
3  import { BrowserRouter, Routes, Route } from "react-router-dom";
4  import ProductList from '../components/Product/ProductList'
5  import ProductAdd from '../components/ProductAdd/ProductAdd';
6  import Error from '../components/Error/Error';
7  import ProductDetail from '../components/ProductDetail/ProductDetail';
8  import DUMMY_PRODUCTS from '../components/Data/Data';
9  import Chat from '../components/SendMessage/Chat';
10
11
12  const App = () => {
13
14    const [products, setProducts] = useState([])
15
16    useEffect(() => {
17      getItem();
18    }, []);
19
20    const getItem = () => {
21      const localData = localStorage.getItem("data") ?? [];
22      if(localData.length===0){
23        localData.push(...DUMMY_PRODUCTS)
24        setProducts(localData);
25      }else{
26        setProducts(JSON.parse(localData));
27      }
28    };
29
30    return (
31      <BrowserRouter>
32        <Navi/>
33        <Routes>
34          <Route path="/addItem" element={<ProductAdd/>} />
35          <Route path="/" element={<ProductList items={products}/>} />
36          <Route path="/chat" element={<Chat/>} />
37          <Route path="/:p_id" element={<ProductDetail items={products}/>} />
38          <Route path="*" element={<Error/>} />
39        </Routes>
40      </BrowserRouter>
41    )
42  }
43
44  export default App
45
```



```
src > js firebase.js > ...
1 import firebase from 'firebase/compat/app'
2 import 'firebase/compat/auth';
3 import 'firebase/compat/firestore';
4
5 const firebaseApp = firebase.initializeApp({
6   apiKey: "AIzaSyD6VhEABNgdC3ssDEusqP67SDT1bFvr1Fk",
7   authDomain: "ikinci-el-esya-adae9.firebaseio.com",
8   projectId: "ikinci-el-esya-adae9",
9   storageBucket: "ikinci-el-esya-adae9.appspot.com",
10  messagingSenderId: "586309497279",
11  appId: "1:586309497279:web:8c09e743f02761eb40710a",
12  measurementId: "G-XBS4VBLFV8"
13 })
14
15 const db = firebaseApp.firestore()
16
17 const auth = firebase.auth()
18
19 export {db, auth}
```

Firestore.js dosyası

Bu dosyada firebase uygulamasında mesajlaşma kısmı için oluşturduğum firestore database'i ve Google ile giriş yapmak için oluşturduğum fonksiyonları çağırdım.

```
src > components > Auth > js SignIn.js > ...
1 import React from 'react'
2 import firebase from 'firebase/compat/app'
3 import {auth} from '../firebase'
4 import { Button } from 'reactstrap';
5
6 function SignIn() {
7   function signInWithGoogle() {
8     const provider = new firebase.auth.GoogleAuthProvider()
9     auth.signInWithPopup(provider)
10   }
11   return (
12     <div>
13       <Button style={{width:'170px',height:'50px'}}
14         onClick={signInWithGoogle}>Google ile giriş yap</Button>
15     </div>
16   )
17 }
18
19 export default SignIn
```

SignIn.js dosyası

Bu dosyada signInWithGoogle fonksiyonu içerisinde Google ile giriş işlemini yaptım ve signInWithPopup fonksiyonu ile butona basıldığında giriş popup'ın çıkmasını sağladım.

```
src > components > Auth > js SignOut.js > default
1 import React from 'react'
2 import { Button } from 'reactstrap';
3 import {auth} from '../firebase'
4
5 function SignOut() {
6   return (
7     <div>
8       <Button style={{width:'100px',height:'50px'}}
9         onClick={() => auth.signOut()}>Çıkış Yap</Button>
10     </div>
11   )
12 }
13
14 export default SignOut
```

SignOut.js dosyası

Bu dosyada Çıkış yap butonunun çalışması için firebase signOut fonksiyonunu kullandım.

```
src > components > Data > js Data.js > ...
1 const DUMMY_PRODUCTS = [
2   {
3     id: "e1",
4     title: "Huawei Matebook",
5     category: "Bilgisayar",
6     imageURL: [
7       "https://cdn.vatanbilgisayar.com/Upload/PRODUCT/huawei/thumb/114724_large.jpg",
8       "https://cdn.vatanbilgisayar.com/Upload/PRODUCT/huawei/thumb/114724-1_large.jpg",
9       "https://cdn.vatanbilgisayar.com/Upload/PRODUCT/huawei/thumb/114724-2_large.jpg",
10     ],
11     description: "D15I 10.Nesil Core i5 10210U-8Gb-256Gb Ssd-15.6inc-W10",
12     price: "7.999",
13     place: "İstanbul/Pendik",
14     username: "Alperen Kapusuz",
15     userGmail: "sefalperen21@gmail.com",
16     userPhoto: "https://i.pinimg.com/736x/25/23/8a/25238a626939a5c2394fae5d1ff7c0c1.jpg",
17   },
18 ]
```

Data.js dosyası

Bu dosyada DUMMY_PRODUCTS adlı bir obje oluşturdum bu objede;

- id
- title
- category
- imageURL dizi (bu dizide 3 adet resim tutulmaktadır)
- description
- place
- username
- userGmail
- userPhoto

Bilgileri tutulmaktadır.

```
src > components > Error > js Error.js > ...
1 import React from "react";
2 import { Link } from "react-router-dom";
3
4 const Error = () => {
5   return (
6     <div>
7       <div role="alert">
8         <h1> Böyle bir sayfa bulunamadı!!! Error404</h1>
9         <Link to="/">
10           anasayfa'ya git
11         </Link>
12       </div>
13     </div>
14   );
15 };
16
17 export default Error;
18
```

Error.js dosyası

src > components > Navbar > Navi.js > ...

```
1  import React from "react";
2  import {
3    Navbar,
4    NavbarBrand,
5    Collapse,
6    NavItem,
7    NavLink,
8    Nav,
9  } from "reactstrap";
10 import SignIn from "../Auth/SignIn";
11 import SignOut from "../Auth/SignOut";
12 import "./Navi.css";
13
14 const Navi = () => {
15   return (
16     <div className="Navbar_page">
17       <Navbar color="light" expand="md" light>
18         <NavbarBrand href="/">ikinci1</NavbarBrand>
19         <Collapse navbar>
20           <Nav className="me-auto" navbar>
21             <div className="Navbar__ads_add">
22               <NavItem className="Navbar__addItems">
23                 <NavLink href="/addItem">Ürün Ekle</NavLink>
24               </NavItem>
25             </div>
26             <div className="Navbar__button">
27               <NavItem>
28                 <NavLink href="#">
29                   <SignOut />
30                 </NavLink>
31               </NavItem>
32               <NavItem>
33                 <NavLink href="#">
34                   <SignIn />
35                 </NavLink>
36               </NavItem>
37             </div>
38           </Nav>
39         </Collapse>
40       </Navbar>
41     </div>
42   );
43 };
44
45 export default Navi;
46
```

Navi.js component

Bu component bizi anasayfaya, ürün ekle sayfasına götürmeyi sağlar, aynı zamanda çıkış yap butonu ve Google ile giriş yap butonları ile auth işlemlerini yapmamızı sağlar.

Komponent içerisinde:

- Reactstrap arayüzü kullandım.
- SignIn ve SignOut componentlerini çağırarak auth işlemlerini uyguladım.

Navi komponentinin görünüşü aşağıdaki gibidir:

ikinci1

Ürün Ekle

Çıkış Yap

Google ile giriş yap

```
src > components > Product > ProductItem.js > ProductItem
```

```
1  import React from "react";
2  import { Link } from "react-router-dom";
3  import {
4    Button,
5    Card,
6    CardBody,
7    CardTitle,
8    CardText,
9  } from "reactstrap";
10 import "./ProductItem.css";
11
12
13
14 const ProductItem = (props) => {
15
16   return (
17     <div>
18       <Card className="ProductItem_card" body outline>
19         <CardBody className="ProductItem_cardBody">
20           <img
21             className="ProductItem_image"
22             src={props.imageURL[0]}
23             alt="Card image cap"
24           />
25           <div className="ProductItem_div1">
26             <CardTitle tag="h4">{props.title}</CardTitle>
27           </div>
28           <div className="ProductItem_div2">
29             <CardText className="mb-2 text-muted" tag="h4">
30               {props.price} TL
31             </CardText>
32             <CardText className="mb-2 text-muted" tag="h6">
33               {props.place}
34             </CardText>
35           </div>
36           <Button color="success">
37             <Link className="ProductItem_button" to={`/${props.id}`}>
38               Detay
39             </Link>
40           </Button>
41         </CardBody>
42       </Card>
43     </div>
44   );
45 };
46
```

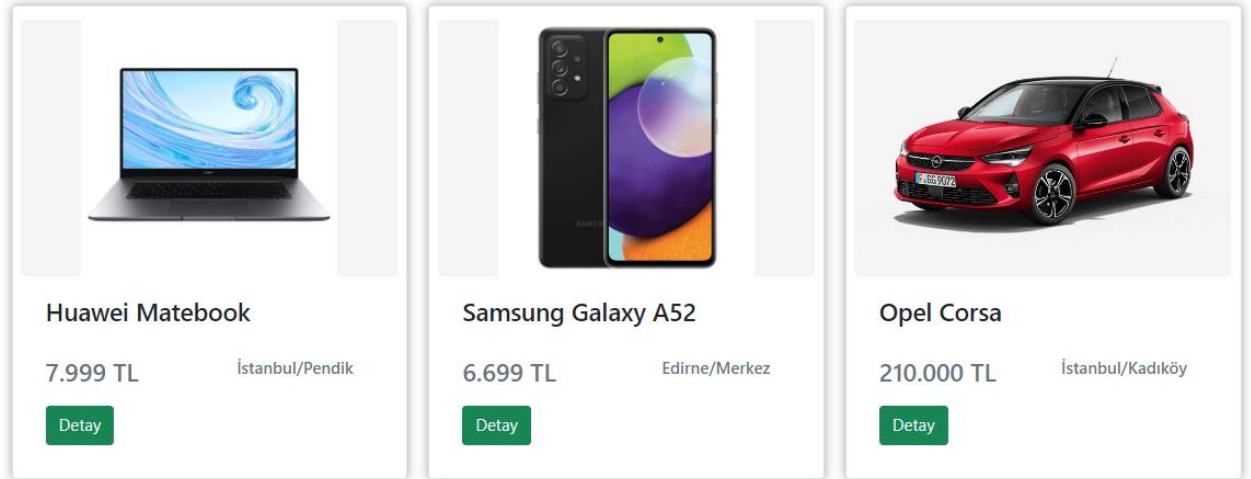
ProductItem.js component

Bu komponent ile Anasayfamızın ana yapısını oluşturan Card görünümleri yaptım.

Komponent içerisinde:

- Reactstrap arayüzü kullandım
- Dom kullanarak Detay butonu ile kullanıcının detay sayfasına gitmesini sağladım
- ProductList'den gelen props ile ürünlerin resimlerini, isimlerini, fiyatını ve konumunu listeledim

ProductList komponentin görünümü aşağıdaki gibidir:



```

6  const ProductList = (props) => {
7    const [searchTitle, setSearchTitle] = useState("");
8    const [searchCategory, setSearchCategory] = useState("");
9    const [searchPlace, setSearchPlace] = useState("");
10
11    if (props.items.length === 0) {
12      return <h2>No expenses found</h2>;
13    }
14
15    return (
16      <div className="productList__page">
17        <div className="productList__search">
18          <Search
19            searchCategory={searchCategory}
20            setSearchCategory={setSearchCategory}
21            searchTitle={searchTitle}
22            setSearchTitle={setSearchTitle}
23            searchPlace={searchPlace}
24            setSearchPlace={setSearchPlace}
25          />
26        </div>
27        <div>
28          {props.items
29            .filter((e) =>
30              e.category
31                .toLocaleLowerCase()
32                .includes(searchCategory.toLocaleLowerCase())
33            )
34            .filter((e) =>
35              e.place
36                .toLocaleLowerCase()
37                .includes(searchPlace.toLocaleLowerCase())
38            )
39            .filter((p) =>
40              p.title
41                .toLocaleLowerCase()
42                .includes(searchTitle.toLocaleLowerCase())
43            )
44            .map((product) => (
45              <ProductItem
46                key={product.id}
47                id={product.id}
48                title={product.title}
49                imageURL={product.imageURL}
50                price={product.price}
51                place={product.place}
52              />
53            ))}
54        </div>
55      </div>
56    );
57  };
58
59  export default ProductList;
60

```

ProductList.js component

Bu komponent'de filtreleme işlemlerini ve ProductItem'a ürünleri listelemesi için propslar gönderdim

Komponent içerisinde:

- searchTitle, searchCategory, searchPlace useState'lerini filter fonksiyonunda kullandım ve Search komponentine gönderdim
- Filter fonksiyonu içerisinde kullanıcının aramalarını küçük harflerle yazma olanağını devreye katarak toLocaleLowerCase fonksiyonu kullandım.
- Map fonksiyonu ile gelen props'u mapleme işlemine tabii tuttum ve Product item'a sırası ile gitmesini sağladım

src > components > Search > Search.js > ...

```
4  const Search = ({
5    searchCategory,
6    setSearchCategory,
7    searchPlace,
8    setSearchPlace,
9    searchTitle,
10   setSearchTitle,
11 }) => {
12   return (
13     <div className="Search__page">
14       <form className="Search__category_div">
15         <label>Category filter</label>
16         <input
17           name="search"
18           list="category"
19           value={searchCategory}
20           className="form-control me-sm-2"
21           type="text"
22           placeholder="Category"
23           onChange={(e) => setSearchCategory(e.target.value)}
24         />
25         <datalist id="category">
26           <option value="Bilgisayar" />
27           <option value="Telefon" />
28           <option value="Araba" />
29           <option value="Beyaz eşya" />
30           <option value="Spor" />
31           <option value="Akseuar" />
32           <option value="Kitap" />
33           <option value="Diğer" />
34         </datalist>
35       </form>
36       <div className="Search__place_div">
37         <label>Place filter</label>
38         <input
39           name="search"
40           value={searchPlace}
41           className="form-control me-sm-2"
42           type="text"
43           placeholder="Place"
44           onChange={(e) => setSearchPlace(e.target.value)}
45         />
46       </div>
47       <div className="Search__title_div">
48         <label>Search</label>
49         <input
50           name="search"
51           value={searchTitle}
52           className="form-control me-sm-2"
53           type="text"
54           placeholder="iPhone"
55           onChange={(e) => setSearchTitle(e.target.value)}
56         />
57       </div>
58     </div>
59   );
```

Search.js component

Bu komponent’de ürünleri filtreleme için kullanılan inputları yaptım

Komponent içerisinde:

- Form yapısı kurdum bu yapının ilk kısmı kategori aramasıdır. Kategori aramasında optionlar ekleyerek kullanıcıya kolaylık sağladım
- İkinci kısmı konum aramasıdır bu kısımda kullanıcının girmiş olduğu konuma göre ürünler listelenir
- Üçüncü kısım ise ürün adı aramasıdır. Kullanıcının girmiş olduğu isme göre ürünler listelenir.

Search Komponent görünümü aşağıdaki gibidir.

Category filter

Place filter

Search

ProductDetail.js component

```
src > components > ProductDetail > ProductDetail.js > ...
1 import React, { useEffect, useState } from "react";
2 import { useParams, Link } from "react-router-dom";
3 import { Button, UncontrolledCarousel } from "reactstrap";
4 import { auth } from "../../firebase";
5 import "../ProductDetail.css";
6 import { useAuthState } from "react-firebase-hooks/auth";
7
8 const ProductDetail = (props) => {
9   const { p_id } = useParams();
10   const [productDetail, setProductDetail] = useState("");
11   const [productDetailImage, setProductDetailImage] = useState([]);
12   const [user] = useAuthState(auth);
13
14   const items = [
15     {
16       src: productDetailImage[0],
17     },
18     {
19       src: productDetailImage[1],
20     },
21     {
22       src: productDetailImage[2],
23     },
24   ];
25
26   useEffect(() => {
27     for (let i = 0; i < props.items.length; i++) {
28       if (props.items[i].id === p_id) {
29         setProductDetail(props.items[i]);
30       }
31     }
32
33     for (let j = 0; j < props.items.length; j++) {
34       if (props.items[j].id === p_id) {
35         setProductDetailImage(props.items[j].imageURL);
36       }
37     }
38   }, []);
```

```
41 return (
42   <div className="ProductDetail_page">
43     <user ? (
44       <div>
45         <div className="ProductDetail_detail">
46           <div>
47             <UncontrolledCarousel
48               indicators={false}
49               className="ProductDetail_image"
50               items={items}
51             />
52           </div>
53           <div>
54             <div className="ProductDetail_userInfo">
55               <div className="ProductDetail_userPhoto">
56                 <img src={productDetail.userPhoto} alt="user photo" />
57               </div>
58               <div className="ProductDetail_username-gmail">
59                 <p>{productDetail.username}</p>
60                 <p>{productDetail.userGmail}</p>
61               </div>
62             </div>
63             <div>
64               <Button className="ProductDetail_chat-button" color="success">
65                 <Link className="ProductDetail_link" to="/chat">
66                   Mesaj Gönder
67                 </Link>
68               </Button>
69             </div>
70             <div className="ProductDetail_title">
71               <p>{productDetail.title}</p>
72             </div>
73             <div className="ProductDetail_cp">
74               <p>{productDetail.category}</p>
75               <p>{productDetail.place}</p>
76             </div>
77             <div className="ProductDetail_price">
78               <p>Price: ₺{productDetail.price}</p>
79             </div>
80             <Button className="ProductDetail_button" color="secondary">
81               <Link className="ProductDetail_link" to="/">
82                 Anasayfaya dön
83               </Link>
84             </Button>
85           </div>
86         </div>
87         <div className="ProductDetail_description">
88           <p>{productDetail.description}</p>
89         </div>
90       </div>
91     ) : (
92       <div className="Chat_signIn">
93         <p>Lütfen Giriş yapınız</p>
94       </div>
95     )
96   </div>
```

Bu komponent'de ürün detay sayfasını oluşturdum:

- Öncelikle useParams fonksiyonu ile ürünün id'sini olarak her ürün için kullanıcının linke gitmesini sağladım.
- İkinci olarak firebase auth fonksiyonunu çağırdım. Bu fonksiyon ile kullanıcı eğer uygulamaya giriş yapmadıysa karşısına Lütfen giriş yapınız yazısı çıkacaktır, giriş yapıldı ise ürün detay sayfası gelecektir.
- useEffect içerisinde propstan gelen ürünlerden seçili olanı for ve if yapıları kullanarak buldum ve bu ürünü productDetail state'i içerisinde attım. Bu state ile seçili olan ürünü detay sayfasında gösterdim
- const items dizisi oluşturduğum Carousel için ürünün resimlerini içermektedir. Carousel buradan resimleri alır ve slayt şeklinde kullanıcıya gösterir.
- ProductDetail componenti görünümü 5. sayfada gösterilmiştir.

```

1 import React, { useState, useEffect, useRef } from "react";
2 import { auth, db } from "../firebase";
3 import SendMessage from "./SendMessage";
4 import "./Chat.css";
5
6 function Chat() {
7   const scroll = useRef();
8   const [messages, setMessages] = useState([]);
9
10  useEffect(() => {
11    db.collection("messages")
12      .orderBy("createdAt")
13      .limit(50)
14      .onSnapshot((snapshot) => {
15        setMessages(snapshot.docs.map((doc) => doc.data()));
16      });
17  }, []);
18
19  return (
20    <div>
21      <div className="page">
22        <div>
23          <SendMessage scroll={scroll} />
24        </div>
25        <div className="msgs">
26          {messages.map(({ id, text, photoURL, uid }) => (
27            <div>
28              <div
29                key={id}
30                className={`msg ${
31                  uid === auth.currentUser.uid ? "sent" : "received"
32                }`}
33              >
34                <img src={photoURL} alt="" />
35                <p>{text}</p>
36              </div>
37            </div>
38          ))}
39        </div>
40        <span ref={scroll}></span>
41      </div>
42    </div>
43  );
44 }
45
46 export default Chat;

```

Chat.js component

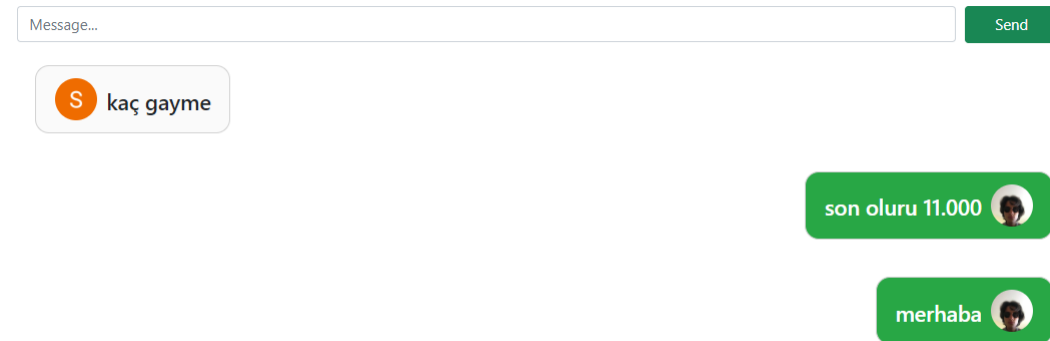
Bu komponent'de firebase kullanarak satıcı ile alıcı arası mesajlaşma kısmını yaptım.

Mesajlaşmalar public bir şekilde yapılmaktadır

Komponent içerisinde:

- Scroll kullandım
- Messages state'i tanımladım. UseEffect içerisinde firebase uygulaması içerisinde kullanıcının mesajlarının gideceği bir database oluşturdum. Database deki data'yı messages state'i içerisine aktardım
- Return içerisinde messages objesini map ettim. Burada mesajın alıcı mı yoksa satıcı mı tarafında gönderildiği belirleniyor

Komponent aşağıdaki gibi görünmektedir.




```

src > components > SendMessage >  SendMessage.js > ...
1  import React, { useState } from "react";
2  import { db, auth } from "../../firebase";
3  import firebase from "firebase/compat/app";
4  import { Input, Button } from "reactstrap";
5  import "./SendMessage.css";
6
7  function SendMessage({ scroll }) {
8    const [msg, setMsg] = useState("");
9
10   async function sendMessage(e) {
11     e.preventDefault();
12     const { uid, photoURL } = auth.currentUser;
13
14     await db.collection("messages").add({
15       text: msg,
16       photoURL,
17       uid,
18       createdAt: firebase.firestore.FieldValue.serverTimestamp(),
19     });
20     setMsg("");
21     scroll.current.scrollToView({ behavior: "smooth" });
22   }
23   return (
24     <div>
25       <form className="sendMessage__form" onSubmit={sendMessage}>
26         <Input
27           value={msg}
28           onChange={(e) => setMsg(e.target.value)}
29           placeholder="Message..."
30         />
31         <Button type="submit" color="success">
32           Send
33         </Button>
34       </form>
35     </div>
36   );
37 }
38
39 export default SendMessage;
40

```

SendMessage.js component

Bu komponent’de firebase datasına kullanıcıdan alınan mesajın gönderilmesi işlemini yaptım.

- SendMessage fonksiyonu içerisinde firestore database’ ine kullanıcının mesajları eklenir.
- Return içerisinde reactstrap arayüzü kullandım. Burada Input ‘dan kullanıcının attığı mesaj alınarak sendMessage fonksiyonuna gönderilir

Komponent görünümü aşağıdaki gibidir.


```
const ProductAdd = () => {
  const [data, setData] = useState([]);
  const [user] = useAuthState(auth);
  const [username, setUsername] = useState("");
  const [userGmail, setUserGmail] = useState("");
  const [userPhoto, setUserPhoto] = useState("");
  const [form, setForm] = useState({
    id: Math.random().toString(),
    title: "",
    category: "",
    imageURL: [],
    description: "",
    price: "",
    place: "",
  });
};
```

ProductAdd.js component

Bu komponent içerisinde ürün localStorage içerisinde ürün ekleme işlemlerini yaptım.

- Reactstrap arayüzü kullandım
- Firebase auth işlemi kullandım. Kullanıcı giriş yapmadıya ürün ekleyememek
- Kullandığımız data bilgilerini içeren form adlı bir state oluşturdum. Bu state'i localStorage işlemlerinde kullanacağız

```
24 const saveItem = () => {
25   if (
26     form.title === "" ||
27     form.category === "" ||
28     form.imageURL.length === 0 ||
29     form.description === "" ||
30     form.price === "" ||
31     form.place === ""
32   ) {
33     alert("Tum alanlari doldurunuz");
34     return;
35   }
36
37   const userInfo = auth.currentUser;
38   let uname = form.username;
39   let uGmail = form.userGmail;
40   let uPhoto = form.userPhoto;
41   if (userInfo !== null) {
42     uname = userInfo.displayName;
43     uGmail = userInfo.email;
44     uPhoto = userInfo.photoURL;
45   }
46   setUsername(uname);
47   setUserGmail(uGmail);
48   setUserPhoto(uPhoto);
49
50   data.push({
51     ...form,
52     username: uname,
53     userGmail: uGmail,
54     userPhoto: uPhoto,
55   });
56
57   localStorage.setItem("data", JSON.stringify(data));
58
59   setForm({
60     title: "",
61     category: "",
62     imageURL: ["", "", ""],
63     description: "",
64     price: "",
65     place: "",
66   });
67   console.log(data);
68 };
```

saveItem fonksiyonu içerisinde:

- if içerisinde eğer kullanıcı bilgileri girmediyse alert verecek
- Firebase auth'dan gelen kullanıcı bilgilerini form içerisinde aktardım
- LocalStorage'da oluşturduğum data'ya form'dan gelen bilgileri ve kullanıcı bilgilerini pushladım
- En sonda ise bilgi girilen inputların işlem tamamlandıktan sonra boşalması için form state'inin içeriğini temizledim

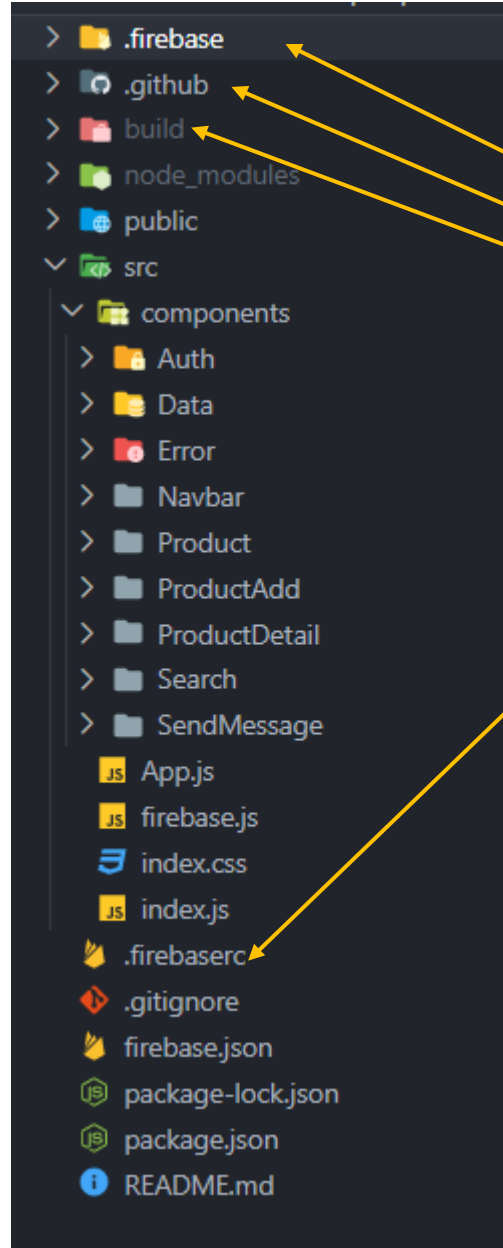
```
70 useEffect(() => {
71   getItem();
72 }, []);
73
74 console.log(username);
75 console.log(userGmail);
76 console.log(userPhoto);
77
78 const getItem = () => {
79   const localData = localStorage.getItem("data") ?? [];
80   if (localData.length === 0) {
81     localData.push(...DUMMY_PRODUCTS);
82     setData(localData);
83   } else {
84     setData(JSON.parse(localData));
85   }
86 };
```

getItem fonksiyonunda app.js de olduğu gibi, oluşturduğum localstorage'dan dataları aldım. LocalStorage boş durumda ise oluşturduğum dummy data'yı local storage'a push ettim, dolu durumda ise localdata parse ederek objeye dönüştürdüm.

```
<FormGroup className="ProductAdd__title">
  <Label>Title</Label>
  <Input
    type="text"
    name="text"
    id="text"
    placeholder="Enter title"
    onChange={(event) => {
      setForm({ ...form, title: event.target.value })
    }}
    value={form.title}
  />
</FormGroup>
```

FormGroup kısmı Input ve Label kısımları bu şekildedir. Form içeriğine input'a girilen veriyi onChange fonksiyonu içerisinde ekledim

Uygulama dosya yapısı



Oluşturulan bu dosyalar
uygulama deploy edildikten
sonra gelmiştir