

Object

프로젝트기반 프론트엔드 웹&앱 SW 개발자 양성과정

전찬혁

-

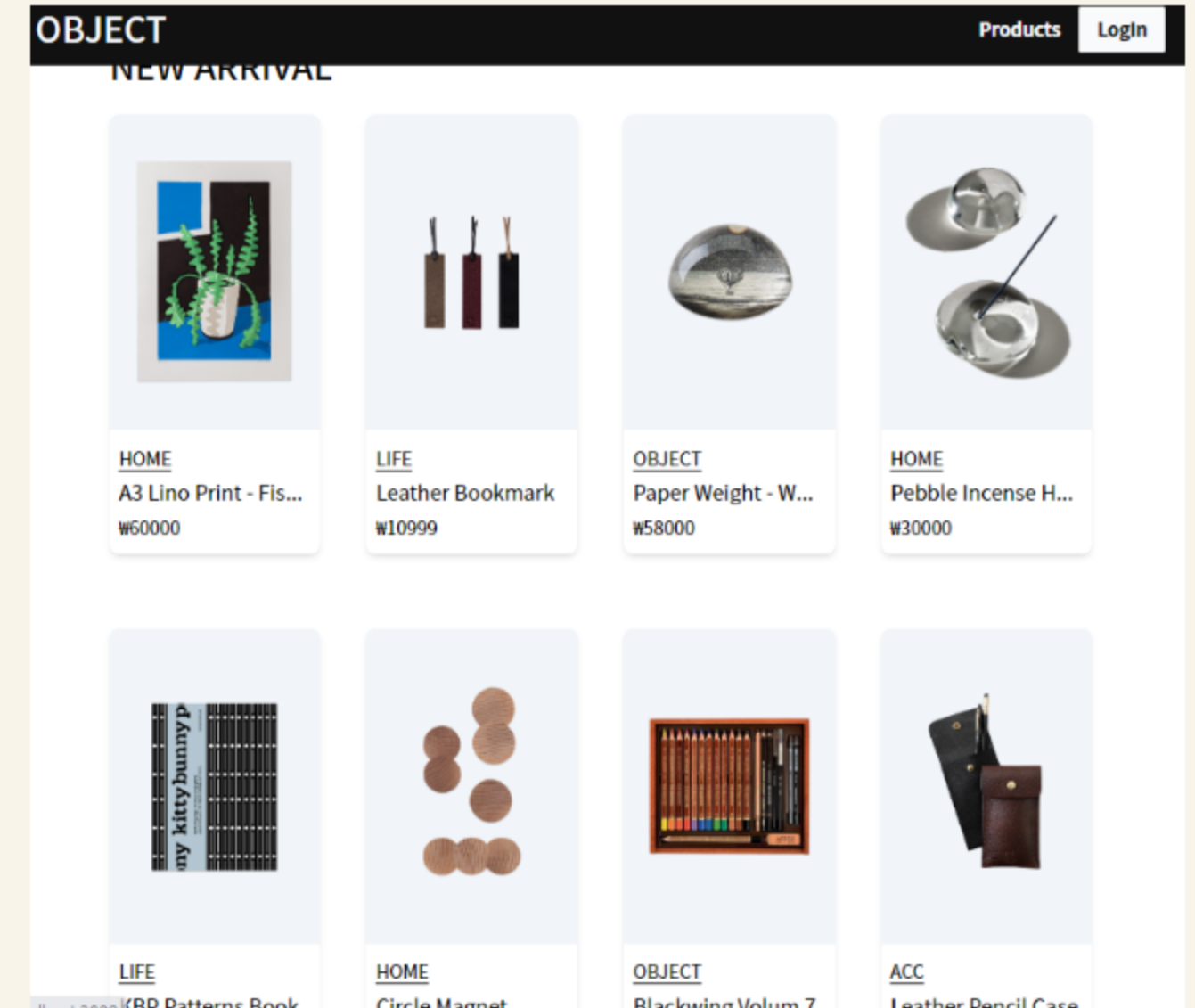
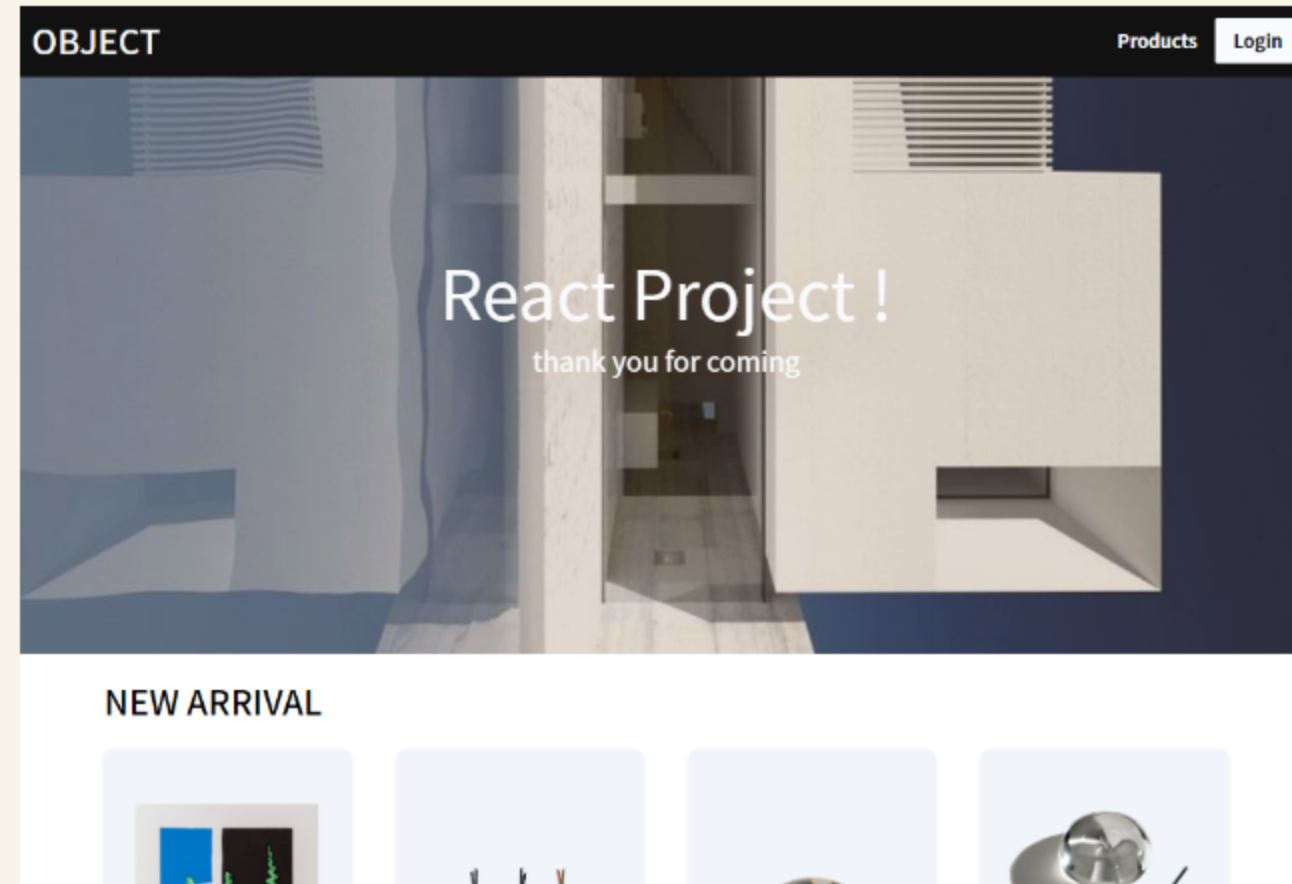
2023.02.10

INDEX

01 Object Project

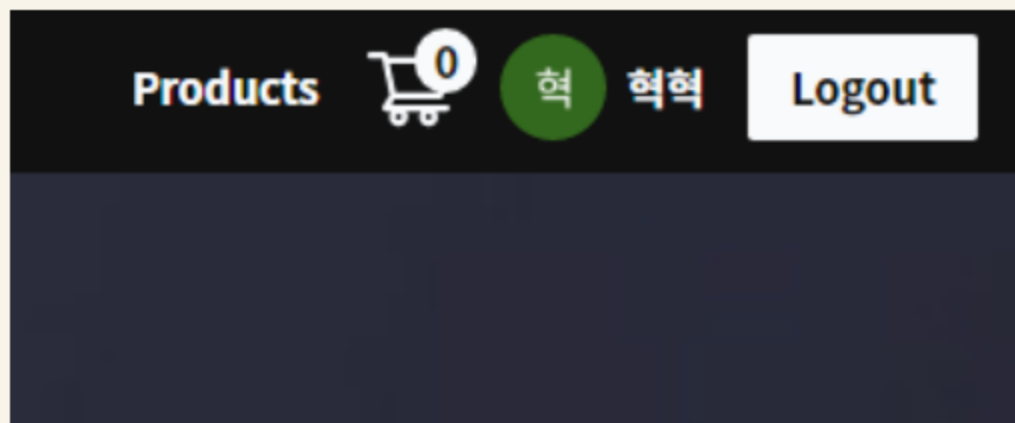
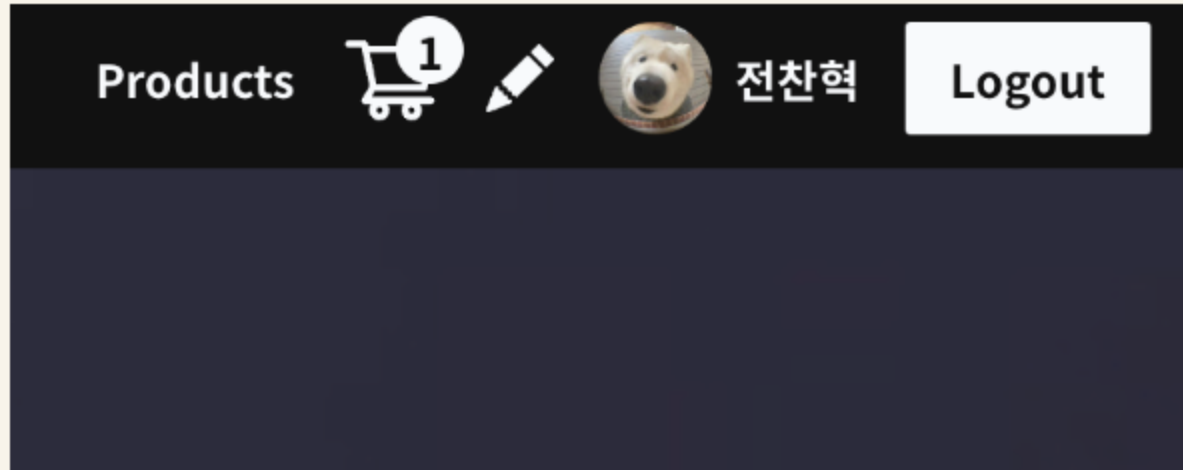
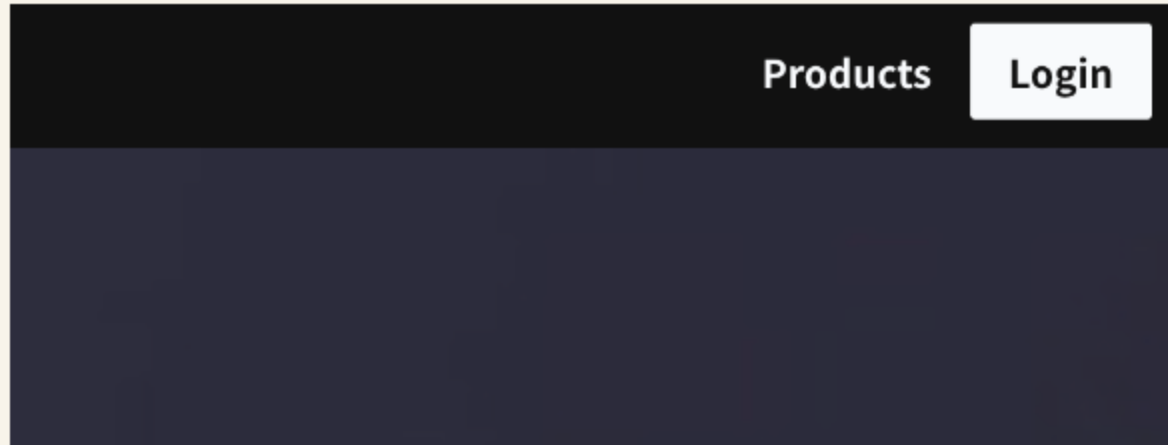
Object Project

- HTML
- CSS
- JavaScript
- React
- Tailwind
- Firebase



Interaction

- 로그인



```
import React from 'react';
import { Link } from 'react-router-dom';
import { BsFillPencilFill } from 'react-icons/bs';
import User from './User';
import { useAuthContext } from '../context/AuthContext';
import CartStatus from './CartStatus';
import Button2 from './ui/Button2';

export default function Navbar() {
  const { user, login, logout } = useAuthContext();
  return (
    <header className='z-10 sticky top-0 flex justify-between px-8 py-3 bg-brand'>
      <Link to='/' className='flex items-center text-3xl text-brand'>
        <h1 className='font-medium text-slate-50'>OBJECT</h1>
      </Link>
      <nav className='flex items-center gap-4 font-semibold'>
        <Link className='text-slate-50' to='/products'>Products</Link>
        {user && (
          <Link to='/carts'>
            <CartStatus />
          </Link>
        )}
        {user && user.isAdmin && (
          <Link to='/products/new' className='text-2xl'>
            <BsFillPencilFill className='text-slate-50' />
          </Link>
        )}
        {user && <User user={user} />}
        {!user && <Button2 text={'Login'} onClick={login} />}
        {user && <Button2 text={'Logout'} onClick={logout} />}
      </nav>
    </header>
  );
}
```

```
import { initializeApp } from 'firebase/app';
import { v4 as uuid } from 'uuid';
import {
  getAuth,
  signInWithPopup,
  GoogleAuthProvider,
  signInOut,
  onAuthStateChanged,
} from 'firebase/auth';
import { getDatabase, ref, set, get, remove } from 'firebase/database';

const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
  databaseURL: process.env.REACT_APP_FIREBASE_DB_URL,
  projectId: process.env.REACT_APP_FIREBASE_PROJECT_ID,
};

const app = initializeApp(firebaseConfig);
const auth = getAuth();
const provider = new GoogleAuthProvider();
const database = getDatabase(app);

export function login() {
  signInWithPopup(auth, provider).catch(console.error);
}

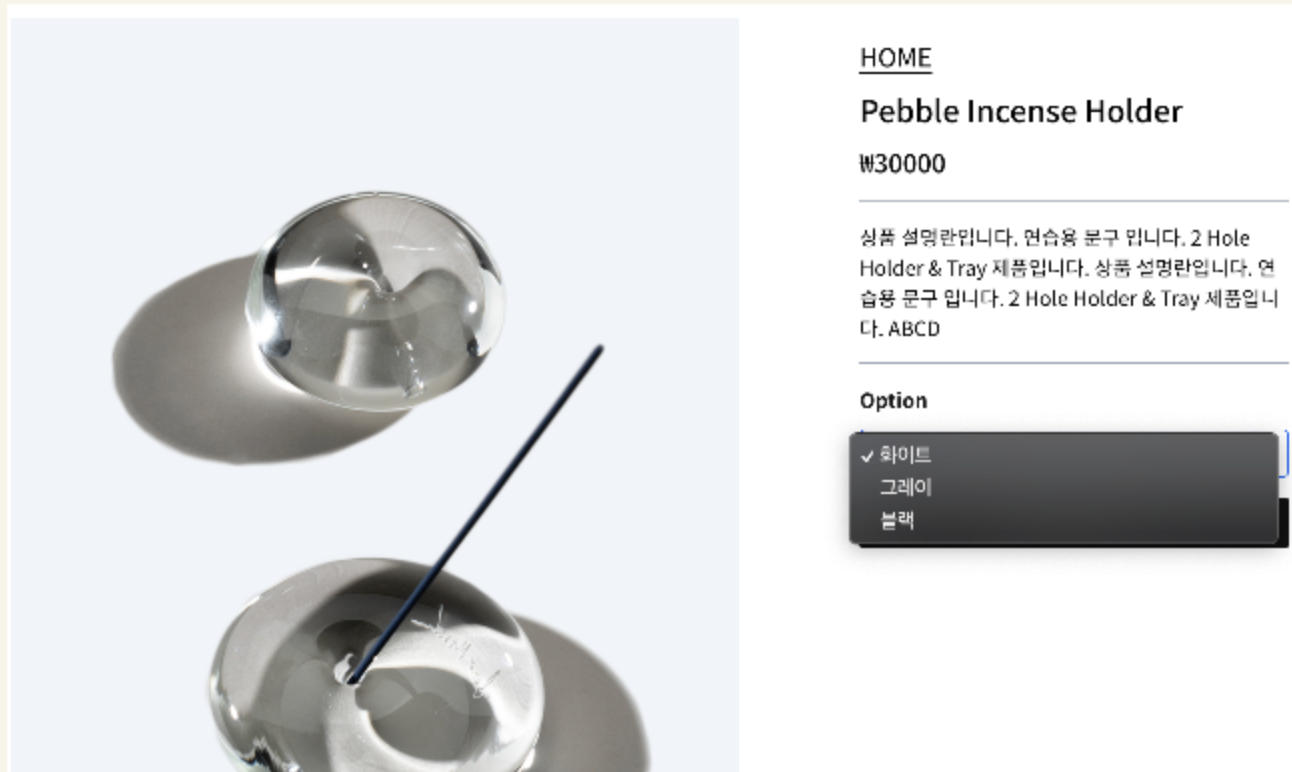
export function logout() {
  signOut(auth).catch(console.error);
}

export function onUserStateChange(callback) {
  onAuthStateChanged(auth, async (user) => {
    const updatedUser = user ? await adminUser(user) : null;
    callback(updatedUser);
  });
}

async function adminUser(user) {
  return get(ref(database, 'admins')) //
    .then((snapshot) => {
      if (snapshot.exists()) {
        const admins = snapshot.val();
        const isAdmin = admins.includes(user.uid);
        return { ...user, isAdmin };
      }
      return user;
    });
}
```

Interaction

- 상품 상세페이지



```
import React, { useState } from 'react';
import { useLocation } from 'react-router-dom';
import { addOrUpdateToCart } from '../api/firebase';
import Button from '../components/ui/Button';
import { useAuthContext } from '../context/AuthContext';

export default function ProductDetail() {
  const { uid } = useAuthContext();
  const {
    state: {
      product: { id, image, title, description, category, price,
    },
  } = useLocation();
  const [selected, setSelected] = useState(options[0]);
  const handleSelect = (e) => setSelected(e.target.value);
  const handleClick = (e) => {
    const product = { id, image, title, price, option: selected,
      addOrUpdateToCart(uid, product);
  };

  return (
    <section className='flex flex-col gap-x-20 justify-center'>
      <img className='h-auto max-w-xl px-4 basis-7/12 bg-slate' />
      <div className='w-full basis-5/12 flex flex-col p-4'>
        <p className='text-xl underline underline-offset-4'>{title}</p>
        <h2 className='text-2xl font-medium py-3'>{title}</h2>
        <p className='text-xl font-medium pb-4 border-b border-gray-400'>₩{price}</p>
        <p className='py-4 text-md border-b border-gray-400'>{description}</p>
        <label className='text-brand font-bold pt-4 pb-3'>Option</label>
        <select
          id='select'
          className='form-select appearance-none block w-full px-3 py-1.5 text-base font-normal text-gray-700 bg-white bg-clip-padding bg-no-repeat border border-solid border-gray-400'>
            {options.map(option => (
              <option
                key={option.id}
                value={option.value}
                selected={option.value === selected}
              >{option.value}</option>
            ))}
          </select>
      </div>
    </section>
  );
}
```

```
async function adminUser(user) {
  return get(ref(database, 'admins')) //
    .then((snapshot) => {
      if (snapshot.exists()) {
        const admins = snapshot.val();
        const isAdmin = admins.includes(user.uid);
        return { ...user, isAdmin };
      }
      return user;
    });
}

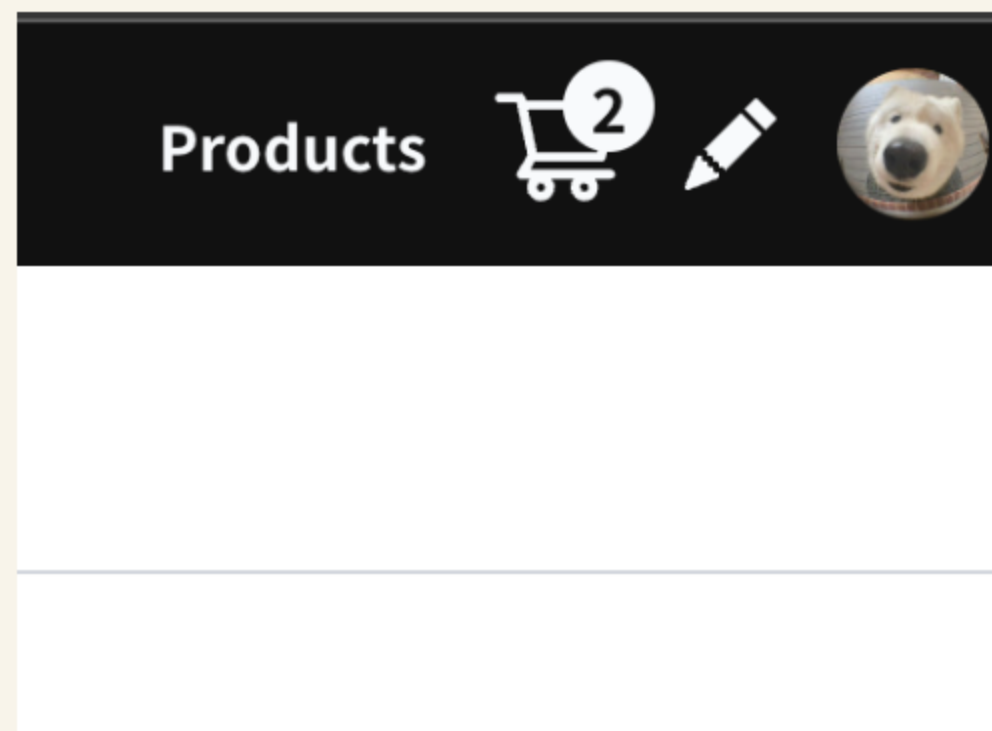
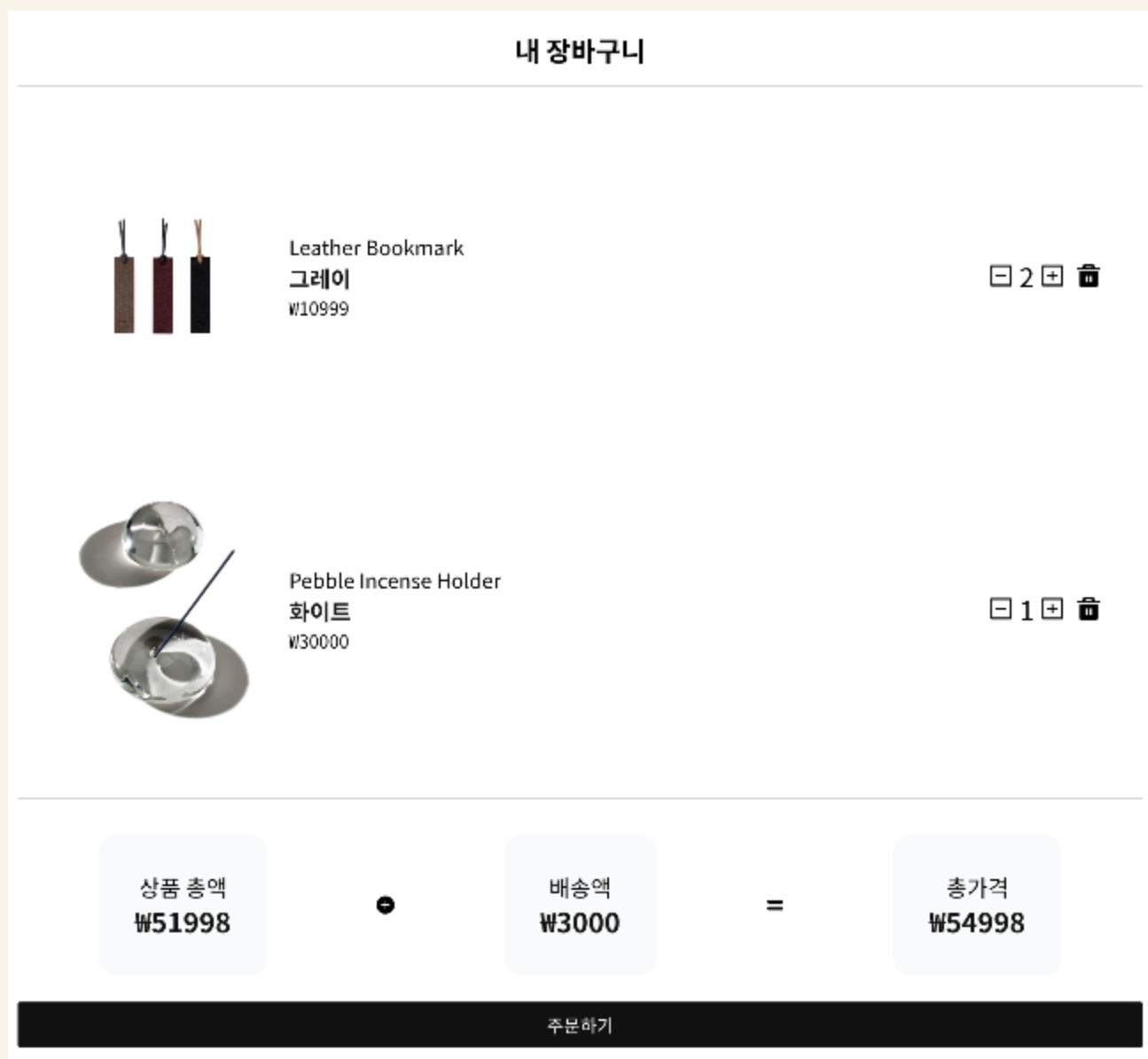
export async function addNewProduct(product, image) {
  const id = uuid();
  return set(ref(database, `products/${id}`), {
    ...product,
    id,
    price: parseInt(product.price),
    image,
    options: product.options.split(','),
  });
}

export async function getProducts() {
  return get(ref(database, 'products')).then((snapshot) => {
    if (snapshot.exists()) {
      return Object.values(snapshot.val());
    }
    return [];
  });
}

export async function getCart(userId) {
  return get(ref(database, `carts/${userId}`)) //
    .then((snapshot) => {
      const items = snapshot.val() || {};
      return Object.values(items);
    });
}
```

Interaction

- 장바구니



```
import React from 'react';
import { getCart } from '../api/firebase';
import { useQuery } from '@tanstack/react-query';
import { useAuthContext } from '../context/AuthContext';
import CartItem from '../components/CartItem';
import { BsFillPlusCircleFill } from 'react-icons/bs';
import { FaEquals } from 'react-icons/fa';
import PriceCard from '../components/PriceCard';
import Button from '../components/ui/Button';

const SHIPPING = 3000;

export default function MyCart() {
  const { uid } = useAuthContext();
  const { isLoading, data: products } = useQuery(['carts', () => getCart(uid)]);

  if (isLoading) return <p>Loading...</p>;

  const hasProducts = products && products.length > 0;
  const totalPrice =
    products &&
    products.reduce(
      (prev, current) => prev + parseInt(current.price) * current.quantity,
      0
    );

  return (
    <section className='p-8 flex flex-col'>
      <p className='text-2xl text-center font-bold pb-4 border-b border-gray-200'>내 장바구니</p>
      </p>
      {!hasProducts && <p>장바구니에 상품이 없습니다. 열심히 쇼핑해 주세요!</p>}
      {hasProducts && (
        <ul className='border-b border-gray-300 mb-8 p-4 px-8'>
          {products &&
            products.map((product) => (
              <CartItem key={product.id} product={product} uid={uid} />
            ))}
        </ul>
        <div className='flex justify-between items-center mb-6 px-2'>
          <PriceCard text='상품 총액' price={totalPrice} />
          <BsFillPlusCircleFill className='shrink-0' />
          <PriceCard text='배송액' price={SHIPPING} />
          <FaEquals className='shrink-0' />
          <PriceCard text='총가격' price={totalPrice + SHIPPING} />
        </div>
        <Button text='주문하기' />
      )}
    </section>
  );
}
```


Interaction

• 상품등록

상품등록



상품 이미지

파일 선택 b0685eb64deb4...4990f9a1e3.png

상품명

ABC

가격

190000

카테고리

HOME

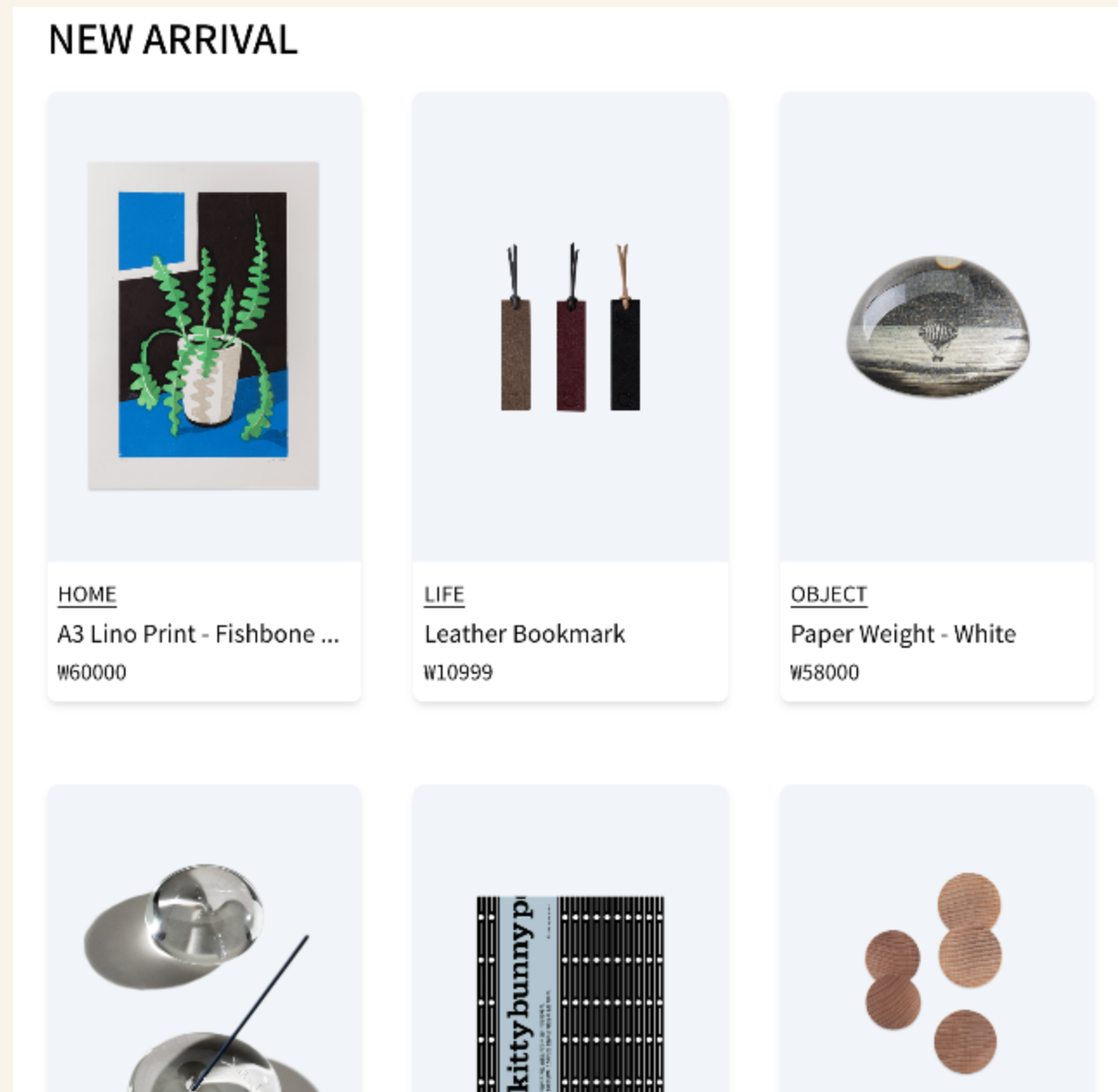
상품 설명

상품 설명란입니다. ABC

옵션

화이트, 블랙

제품 등록하기



```
import React, { useState } from 'react';
import { addNewProduct } from '../api/firebase';
import { uploadImage } from '../api/uploader';
import Button from '../components/ui/Button';

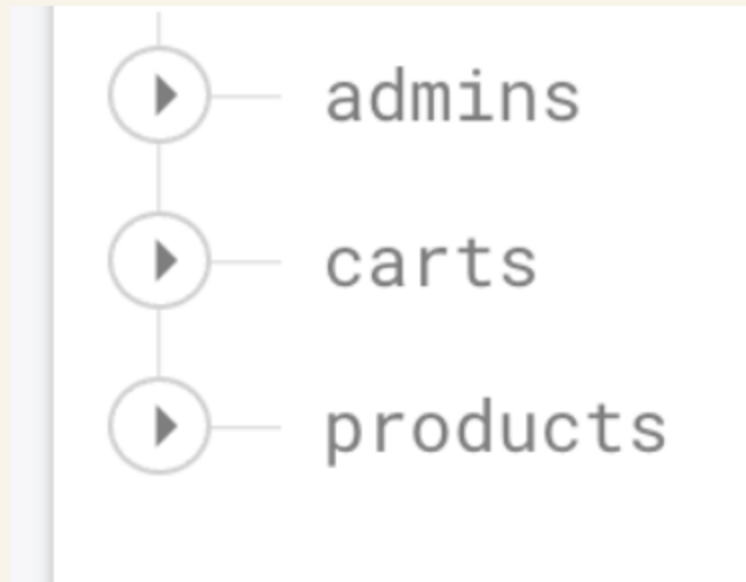
export default function NewProduct() {
  const [product, setProduct] = useState({});
  const [file, setFile] = useState();
  const [isUploading, setIsUploading] = useState(false);
  const [success, setSuccess] = useState();

  const handleChange = (e) => {
    const { name, value, files } = e.target;
    if (name === 'file') {
      setFile(files[0]);
      console.log(files[0]);
      return;
    }
    setProduct((product) => ({ ...product, [name]: value }));
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    setIsUploading(true);
    uploadImage(file) //
      .then((url) => {
        addNewProduct(product, url) //
          .then(() => {
            setSuccess('성공적으로 제품이 추가되었습니다.');
```

Interaction

- 파이어베이스 DB



```
id: "51f308b8-225  
image: "http://res.  
option: "그레이"  
price: 10999  
quantity: 2  
title: "Leather B
```

```
category: "HOME"  
description: "상품 설명란  
id: "3a9cae95-7324-4e87-  
image: "http://res.cloudina  
options  
  0: "화이트"  
  1: " 그레이"  
  2: " 블랙"  
price: 60000  
title: "A3 Lino Print - Fish
```

```
import { initializeApp } from 'firebase/app';  
import { v4 as uuid } from 'uuid';  
import {  
  getAuth,  
  signInWithPopup,  
  GoogleAuthProvider,  
  signOut,  
  onAuthStateChanged,  
} from 'firebase/auth';  
import { getDatabase, ref, set, get, remove } from 'firebase/database';  
  
const firebaseConfig = {  
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,  
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_D  
  databaseURL: process.env.REACT_APP_FIREBASE_DB_UP  
  projectId: process.env.REACT_APP_FIREBASE_PROJECT  
};  
  
const app = initializeApp(firebaseConfig);  
const auth = getAuth();  
const provider = new GoogleAuthProvider();  
const database = getDatabase(app);  
  
export function login() {  
  signInWithPopup(auth, provider).catch(console.err  
}  
  
export function logout() {  
  signOut(auth).catch(console.error);  
}  
  
export function onUserStateChange(callback) {  
  onAuthStateChanged(auth, async (user) => {
```


| Object

Object