

ElecAn



ElecAn.

전자기기 판매 사이트 제작

Keyword 02

Keyword 03

01

프로젝트

선택 이유

03

마치며

02

개발 모습

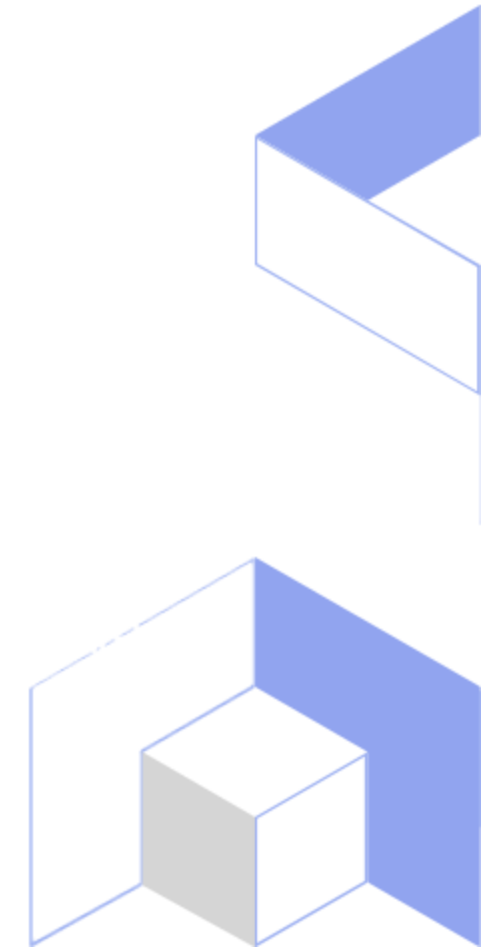
프로젝트

ElecAn

React라는 프로그램을 사용하여 프론트엔드 개발에 좀 더 자기계발을 하기 위해서 개발하였습니다.

React 를 사용해서 제작한 전자기기 판매 사이트를 제작

1. React-bootstrap 을 이용하여 레이아웃 설정
2. 메인 이미지 외 아래 3열로 제품 구성
3. 제품 각각의 설명
4. 장바구니 추가 기능



페이지 구성

image / routes / store(데이터베이스)
폴더 생성

-App.js 가장 많이 사용하는 페이지

image

routes

store

App

App

App.test

data

data2

index

index

logo


main-bg


reportWebVitals

setupTests

store

이름

 Cart

 Detail

페이지 구성

routes

해당 페이지내에 장바구니, 제품 상세 페이지

페이지 구성

App.js

함수 App 을 기본으로 사용하였으며
확인하는데 필요한 내용을 구성

useEffect(localStorage) -> DB구현에 제한이 있어서 관리자 도구에서
데이터베이스를 끌어서 사용

```
function App() {  
  
  let [item , setItem] = useState(data);  
  
  let navigate = useNavigate();  
  
  let obj = {name : 'AnJin'}  
  localStorage.setItem('data',JSON.stringify(obj));  
  let qwe = localStorage.getItem('data');  
  
  useEffect(()=>{  
    | localStorage.setItem('watched',JSON.stringify([]),[]);  
  })  
}
```

페이지 구성

App.js

js 내에서 react-bootstrap 을 import 시켜와서
해당 레이어의 css, 기능 들을 추가

```
import './App.css';
import { createContext, useEffect, useState } from "react";

import { Button , Navbar, Container , Nav} from 'react-bootstrap';
import 이미지 from './image/main_bg.png';
```

```
<div className="App">
  <Navbar bg="dark" variant="dark">
    <Container>
      <Navbar.Brand onClick={() =>{navigate('/')}}>ElecAn</Navbar.Brand>
      <Nav className="me-auto">
        <Nav.Link onClick={() =>{navigate('/')}}>Home</Nav.Link>
        <Nav.Link href="#features">Brand</Nav.Link>
        <Nav.Link onClick={() =>{navigate('/detail')}}>Dart</Nav.Link>
        <Nav.Link onClick={() =>{navigate('/Cart')}}>Cart</Nav.Link>
        <Nav.Link onClick={() =>{navigate('/Event/two')}}>Event</Nav.Link>
      </Nav>
    </Container>
  </Navbar>
</div>
```

ElecAn Home Brand Dart Cart Event



apple
390000



samsung
260000



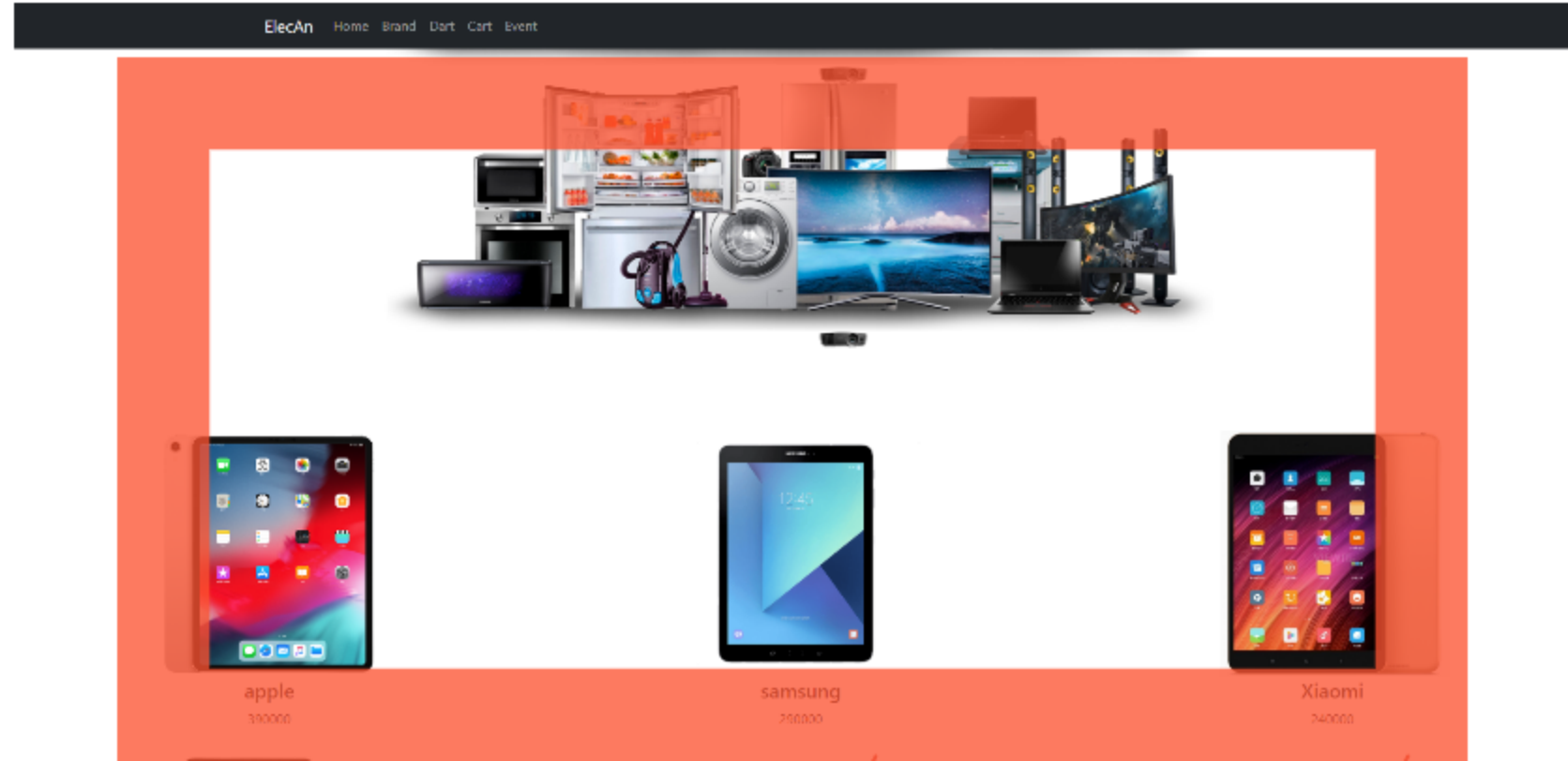
Xiaomi
240000

페이지 구성

App.js

<Route> 사용 하여 페이지 내부 구성

```
<Routes>
  <Route path="/" element={
    <>
      <div className="main-bg"></div>
      <div className="container">
        <div className="row">
          {item.map((a,i)=>{
            return <Card item={item[i]} i={i+1}></Card>
          })}
        </div>
      </div>
    </>
  }/>
  <Route path="/detail/:id" element={<Detail item={item}/>}/>
</Routes>
```




```
JS App.js M JS data.js X JS index.js
src > JS data.js > [0] data
1 let data = [
2   {
3     id : 0,
4     title : "apple",
5     content : "i pad",
6     price : 390000
7   },
8
9   {
10    id : 1,
11    title : "samsung",
12    content : "Galaxy tab",
13    price : 290000
14  },
15
16  {
17    id : 2,
18    title : "Xiaomi",
19    content : "Mi Pad",
20    price : 240000
21  },
22  {
23    id : 3,
24    title : "Lg",
25    content : "G pad",
26    price : 310000
27  },
28
29  {
30    id : 4,
31    title : "상품 준비중",
32    content : "상품 준비중",
33    price : '상품 준비중'
34  },
35
```

페이지 구성

data.js

우선적으로 데이터를 받기 위해
임의 데이터를 js파일로 생성해
export default 를 사용하여 데이터 내보내기

```
    price : '상품 준비중'
  }
]

export default data;
```

```
import { useContext, useEffect, useState } from "react";
import { useParams } from "react-router-dom";
import {Nav} from 'react-bootstrap'
import {addItem} from "../store.js";
import {useDispatch} from "react-redux"
```

```
function Detail(props){
```

```
    let {id} = useParams();
    console.log(id);
    let find_item = props.item.find(x => x.id == id);
    let [count, setCount] = useState(0);
    let [alert, setAlert] = useState(true);
    let [itemTab, setItemTab] = useState(0);
    let dispatch = useDispatch();
    useEffect(()=>{
        let a = setTimeout(()=>{setAlert(false) },2000)
        return () =>{
            clearTimeout(a);
        }
    },[count])
```

```
    console.log('hi');
```

```
    let [fade2 , setFade2] = useState('');
    useEffect(()=>{
```

페이지 구성

detail.js

각 제품의 상세 페이지

상세 페이지 내에서 각자 다른 내용의 탭 구현

```
import { useContext, useEffect, useState } from "react";
import { useParams } from "react-router-dom";
import {Nav} from 'react-bootstrap'
import {addItem} from "../store.js";
import {useDispatch} from "react-redux"
```

```
function Detail(props){
```

```
    let {id} = useParams();
    console.log(id);
    let find_item = props.item.find(x => x.id == id);
    let [count, setCount] = useState(0);
    let [alert, setAlert] = useState(true);
    let [itemTab, setItemTab] = useState(0);
    let dispatch = useDispatch();
    useEffect(()=>{
        let a = setTimeout(()=>{setAlert(false) },2000)
        return () =>{
            clearTimeout(a);
        }
    },[count])
```

```
    console.log('hi');
```

```
    let [fade2 , setFade2] = useState('');
    useEffect(()=>{
```

페이지 구성

detail.js

각 제품의 상세 페이지

상세 페이지 내에서 각자 다른 내용의 탭 구현

```
let [fade2 , setFade2] = useState('');
useEffect(()=>{

    let a = setTimeout(()=>{ setFade2('end')}, 100)
    return ()=>{
        clearTimeout(a);
        setFade2('');
    }
},[])
```

페이지 구성

detail.js

주문하기 버튼 외에 해당 제품의 설명(data.js)을 추가
tabs 내에서 제품의 설명(각기 다른 설명)

```
<div className="row mb-5">
  <div className="col-md-6">
    <img src={process.env.PUBLIC_URL + "/logo"+ find_item[id] + ".png"} width="80%" height="200px"/>
  </div>
  <div className="col-md-6 mt-4">
    <h4 className="pt-5">{props.item[id].title}</h4>
    <p>{find_item.content}</p>
    <p>{find_item.price}원</p>
    <button className="btn btn-danger" onClick={()=>{
      dispatch(addItem ({id : 1, name : 'samsung', count : 1}))
    }}>주문하기</button>
  </div>
</div>
<Nav variant="tabs" defaultActiveKey="link0">
  <Nav.Item>
    <Nav.Link onClick={()=>{setItemTab(0)}}eventKey="link0">Active</Nav.Link>
  </Nav.Item>
  <Nav.Item>
    <Nav.Link onClick={()=>{setItemTab(1)}}eventKey="link1">Option 2</Nav.Link>
  </Nav.Item>
  <Nav.Item>
    <Nav.Link onClick={()=>{setItemTab(2)}}eventKey="link2" >
      3333
    </Nav.Link>
  </Nav.Item>
</Nav>
<TabContent itemTab={itemTab}/>
</div>
```

```

import { configureStore, createSlice } from '@reduxjs/toolkit'
import user from './store/userSlice'

export let { changeName, increase } = user.actions
let cart = createSlice({
  name : 'cart',
  initialState : [
    { id : 0, name : 'ipad', count : 2 },
    { id : 2, name : 'Mi pad', count : 1 }
  ],
  reducers : {
    addCount(state, action) {
      state[action.payload].count++
    },
    addItem(state, action) {
      state.push(action.payload)
    }
  }
})

export let { addCount, addItem } = cart.actions
export default configureStore({
  reducer: {
    user : user.reducer,
    cart : cart.reducer
  }
})

```

페이지 구성

store.js

미리 설정해놓은 데이터를 갖고와 보여줌

```

import {Table} from 'react-bootstrap'
import { useDispatch, useSelector } from 'react-redux'
import { changeName, increase } from "../store.js"
import { addCount } from '../store.js'
function Cart(){

  let state = useSelector((state)=>state)
  let dispatch = useDispatch()

  return(
    <div>

      {state.user.name} {state.user.age}의 장바구니
      { /* <button onClick = {()=>{dispatch(increase())}}>++</button> */ }

      <Table>
        <thead>
          <tr>
            <th>#</th>
            <th>상품명</th>
            <th>수량</th>
            <th>변경하기</th>
          </tr>
        </thead>
        <tbody>
          {
            state.cart.map((a, i)=>
              <tr key={i}>
                <td>{state.cart[i].id}</td>
                <td>{state.cart[i].name}</td>
                <td>{state.cart[i].count}</td>
                <td>
                  <button onClick={()=>{
                    dispatch(addCount(i))
                  }}>+</button>
                </td>
              </tr>
            )
          }
        </tbody>
      </Table>
    </div>
  )
}

```

페이지 구성

cart.js

장바구니 페이지

제품의 상세페이지 내에서 주문하기 버튼 클릭 시
장바구니 페이지에 추가

페이지 구성

```
import { configureStore, createSlice } from '@reduxjs/toolkit'

let user = createSlice({
  name : 'user',
  initialState : { name : 'An', age : 28 },
  reducers : {
    changeName(state){
      state.name = 'park'
    },
    increase(state , action){
      state.age += action.payload
    },
  }
})

export default user
```

userSlice.js

사용자 정보

사용자의 정보를 확인

ElecAn