



2024

BROTHERHOOD

FIFTH PROJECT PROPOSAL

T1 SHOP Website

React

Park Dong-yeol

CONTENTS

01

주제 선정



브레인 스토밍

시장조사

02

기획



페르소나

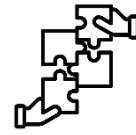
컨셉 설정

와이어 프레임

프로토타입

03

구현화



코드 / 구현화면

04

마무리



소감

Project 일정



2024

BROTHERHOOD



April



01	02	03	04	05	06	07	08	09	10	11
MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU

주제 선정
자료조사

코딩
메인 페이지

12	13	14	15	16	17	18	19	20	21	22
FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON

코딩
메인 페이지

코딩
Detail, Sub, 기능

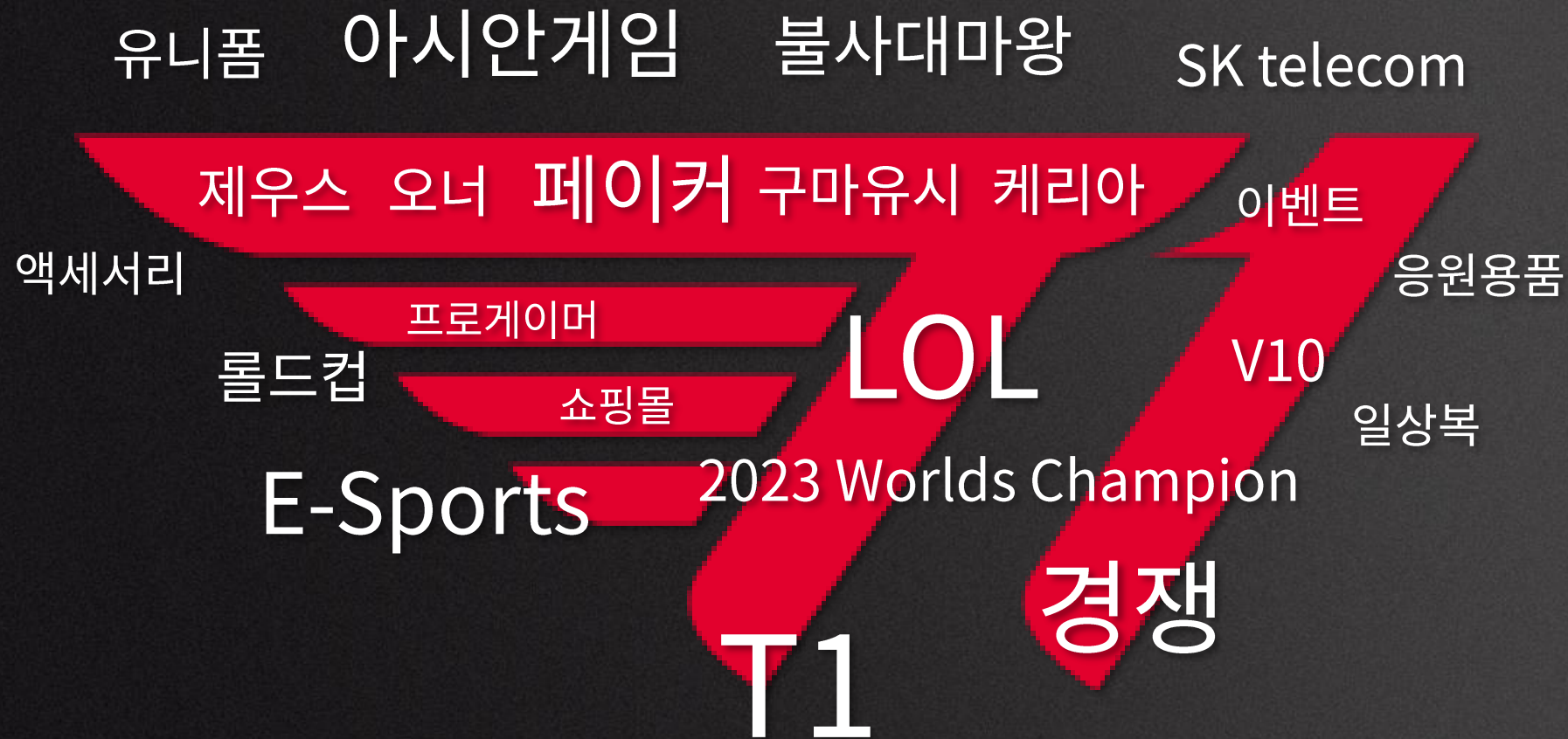
검토 제출

브레인스토밍



2024

BROTHERHOOD





2024

BROTHERHOOD



시장조사 및 전망

시장조사

아시안 게임 종목에 포함되어 'E-Sports'의 위상이 높아지는 현재, **가장 인기 있는 프로게임단**을 꼽으라면 'SK telecom' 산하의 'T1(구, SKT T1)'을 꼽을 수 있을 것이다. 전 세계의 강한 팀들이 모여 펼치는 'World Championship'의 우승컵을 **제일 많이 가진 팀**이자, 국내 프로리그 'LCK'의 우승컵을 **가장 많이** 따낸 팀 역시 T1이다. 그 중에서도 2012년에 데뷔해서 팀의 모든 기록과 함께한 '페이커(이상혁)' 선수를 필두로 국내뿐 아니라 해외에도 많은 팬을 보유한 인기구단 T1은 2023년 한국에서 개최된 '2023 World Championship'을 **우승**함으로써 'World Championship'의 **우승 기록을 4회**로 늘려 타의 추종을 불허하는 **명문 프로게임단**으로의 입지를 한층 더 강화했다.

강력한 게임 실력, 상대를 압도하는 화려한 플레이스타일로 '리그 오브 레전드'의 프로씬에 2012년, 혜성처럼 등장한 '페이커(이상혁)' 선수. 국내 뿐 아니라 국제대회의 우승기록만으로도 타의 추종을 불허하는 그의 행보와 인기는 **프로게이머**라는 직업을 초월해 수많은 사람들에게 영감을 불러일으키고 있다. 광저우 아시안 게임에도 국가대표로 출전하여 금메달을 목에 건 그는 '게임이라도 자신들이 노력하고 경쟁하는 것이 **긍정적인 영감을 불러 일으킨다면 스포츠라고 생각한다**'며, 끊임없는 동기부여와 승부욕으로 자신들을 응원해주는 팬들에게 보답하고자 항상 노력하고 있다. 또 프로게임단의 경우, 특성상 **소속 선수가 자주 바뀌는 경향**이 있으나 T1은 **현재의 로스터를 오랫동안 유지하며 호흡을 맞춰오고 있어** 팬들이 안정적으로 응원을 할 수 있는 환경을 갖추기도 했다.

전망



PROFILE

이름 / 이재완

나이 / 28세

직업 / 스트리머

취미 / 게임

호기심

순발력

내향적

게임 ‘리그 오브 레전드’를 즐겨 플레이한다. 주 포지션은 서포터. 스트리머이기 때문에 주로 방송을 하면서 게임을 플레이한다. 다룰 수 있는 챔피언의 폭이 넓고, 게임을 읽고 분석하는 능력이 일반 사람들 보다 뛰어나 그의 방송을 보러 오는 사람들이 많다. 또 ‘리그 오브 레전드’의 국내 리그와 국제대회가 시작되면 해당 경기들을 중계하며 시청자들과 함께 경기를 지켜본다. T1의 팬이어서 T1의 경기가 있는 날에는 방송을 꼭 켜고 중계방송을 송출한다. T1을 응원하는 마음으로 T1의 유니폼을 비롯하여 응원용품뿐 아니라 일상복으로도 입을 수 있는 의류에 관심이 많다.



PROFILE

이름 / 김은혜

나이 / 32세

직업 / 회사원

취미 / E-Sports시청

열정적

감수성

외향적

게임을 잘 하지는 못하지만 보는 것은 좋아한다. 예전부터 T1의 팬으로 특히 페이커 선수의 팬이다. 경기장에 직접 보러 가는 경우도 있지만 요즘은 일이 바빠서 티켓을 예매하는 것이 어려워 집에서 방송으로 응원하는 편이다. 오랜 기간 한 팀을 응원하면서 많은 선수들이 거쳐가는 것을 보았고, T1이라는 팀이 항상 승승장구하는 것은 아니었기에 응원하기 힘겨운 시기도 있었다. 하지만 결국 꾸준히 버티며 세간의 의심을 보란 듯이 꺾고, 소위 ‘롤드컵’의 우승컵을 들어올리는 페이커 선수의 모습을 보았을 때 그간의 노력들이 결실을 맺는 기분에 감격의 눈물을 흘리기도 했다. 그리고 그런 ‘페이커’선수를 본받아 자신도 더 열심히 살아야겠다는 다짐을 한다.

컨셉 설정



2024

BROTHERHOOD



Typography

일관된 레이아웃에서 다양한 상품정보를 효과적으로 보여주기 위해 'Noto Sans KR(본고딕)' 고딕 폰트와 'Montserrat'를 사용하여 균형감 있는 사이트를 구성했습니다.

Noto Sans KR

높은 가격순정렬

Montserrat

BROTHERHOOD

T1 OFFICIAL UNIFORM

2023 WORLDS CHAMPION

Color System

전체적으로 다양한 색의 사용은 지양하고, T1의 강력한 힘이 느껴지는 강조색과 테마로 꾸몄습니다.

Point Color

#E2012D

C:0% M:100% Y:80% k:11%

Main Color

#000000

C:0% M:0% Y:0% k:100%

Sub Color

#454444

C:0% M:1% Y:1% k:73%

Sub Color

#FFFFFF

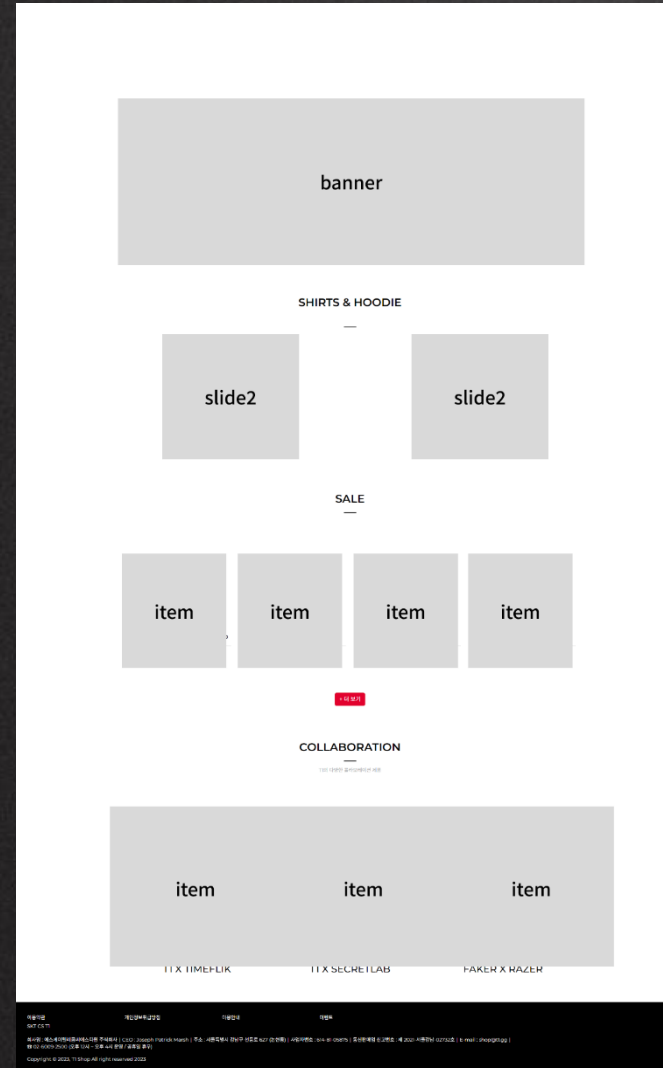
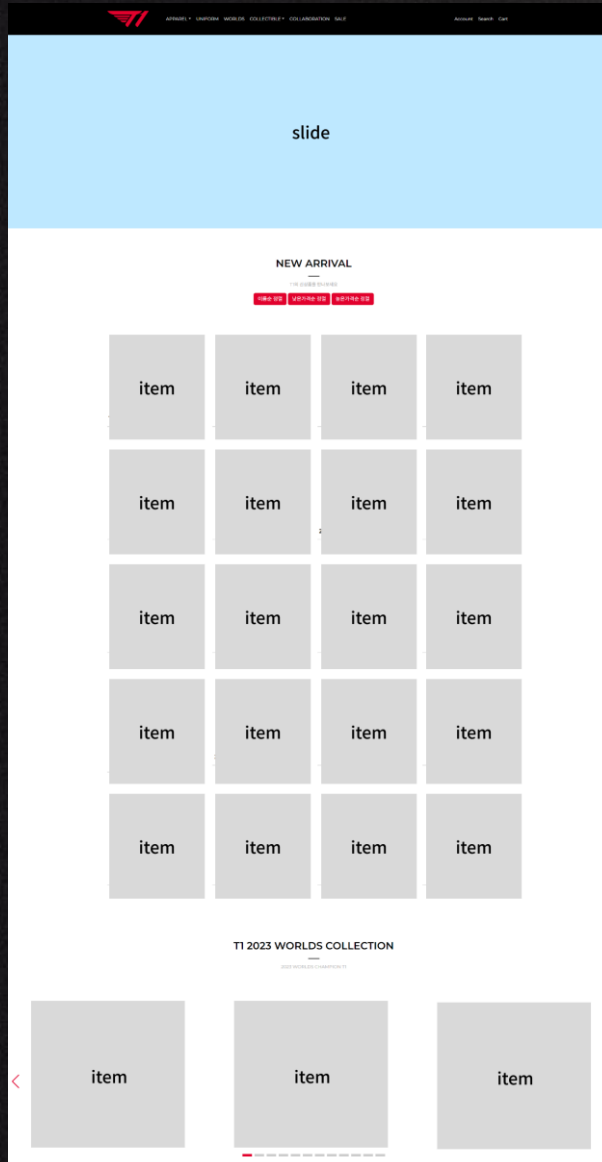
C:0% M:0% Y:0% k:0%

메인페이지 와이어프레임



2024

BROTHERHOOD

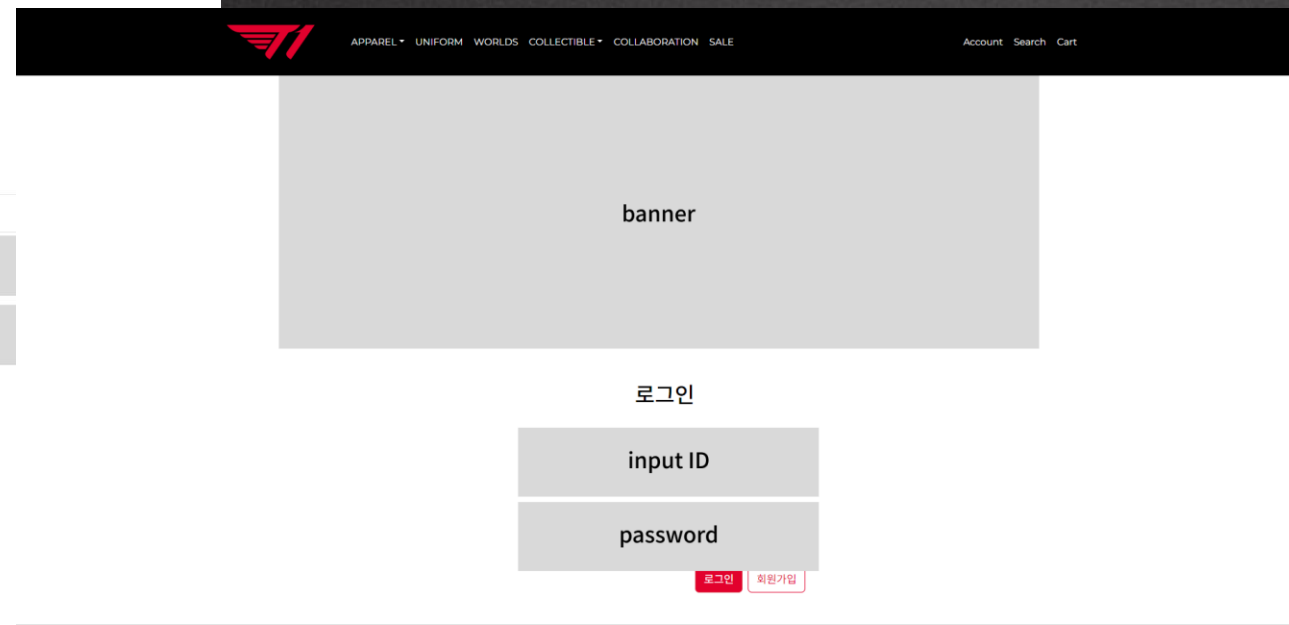
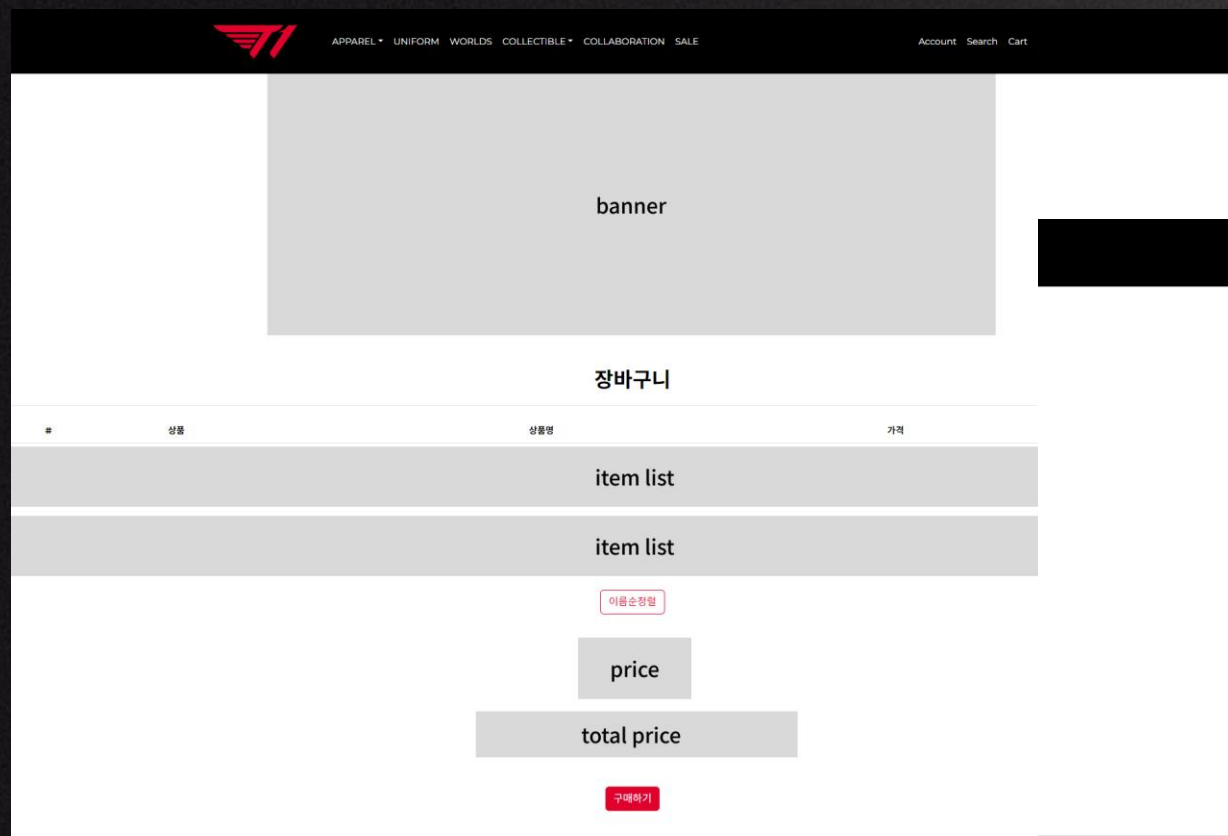


서브페이지 와이어프레임



2024

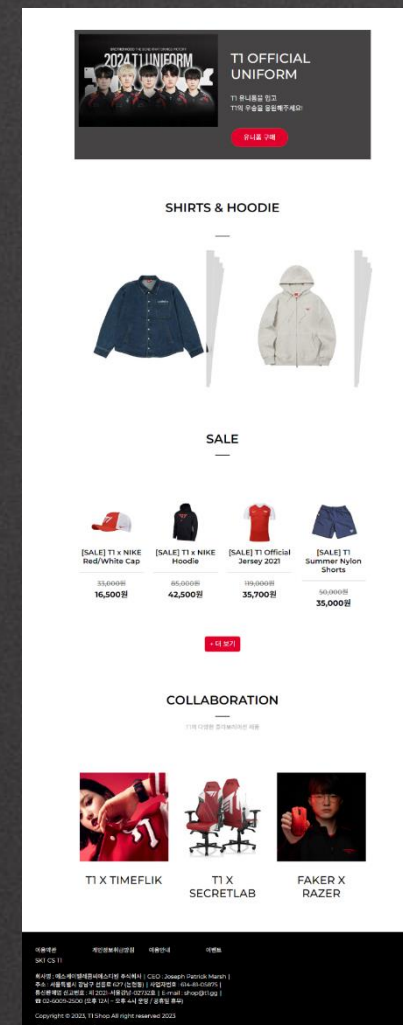
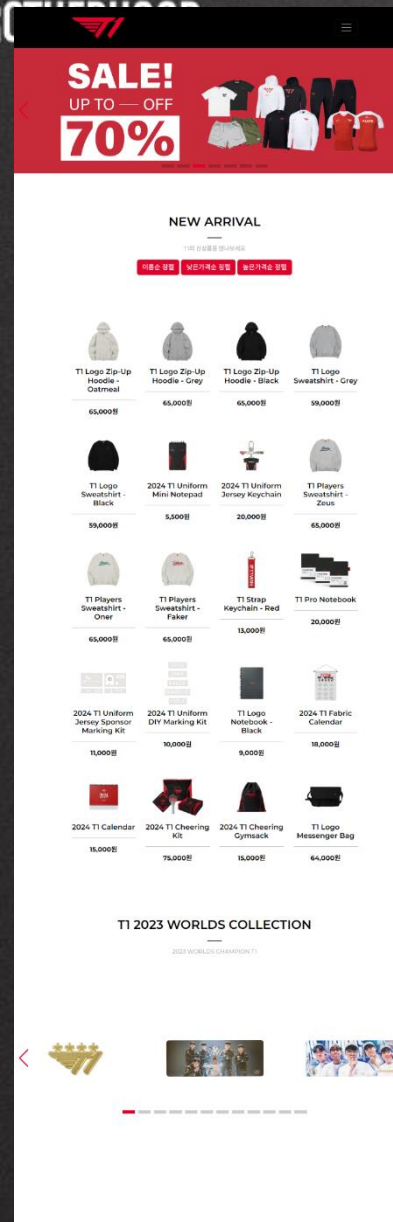
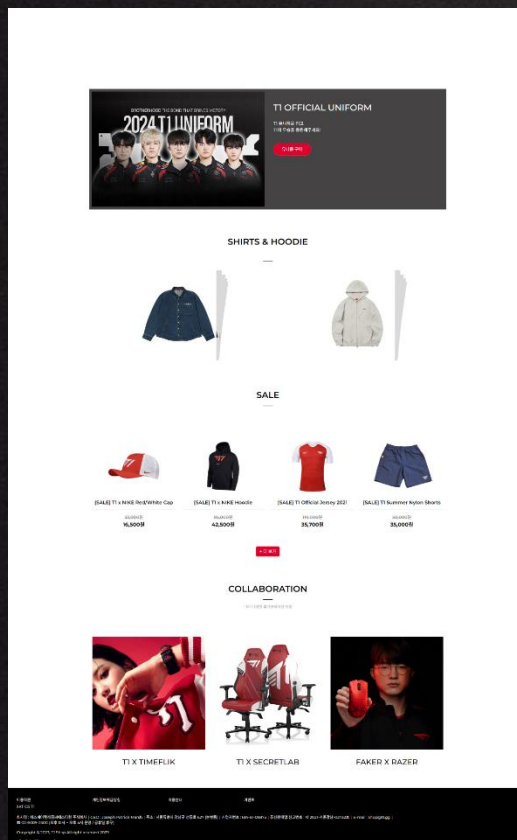
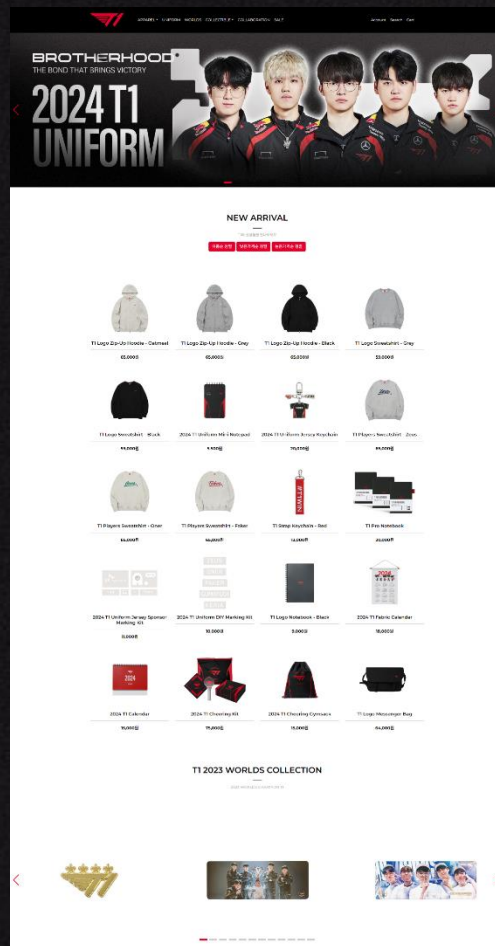
BROTHERHOOD



메인페이지 프로토타입

2024

BROTHERHOOD



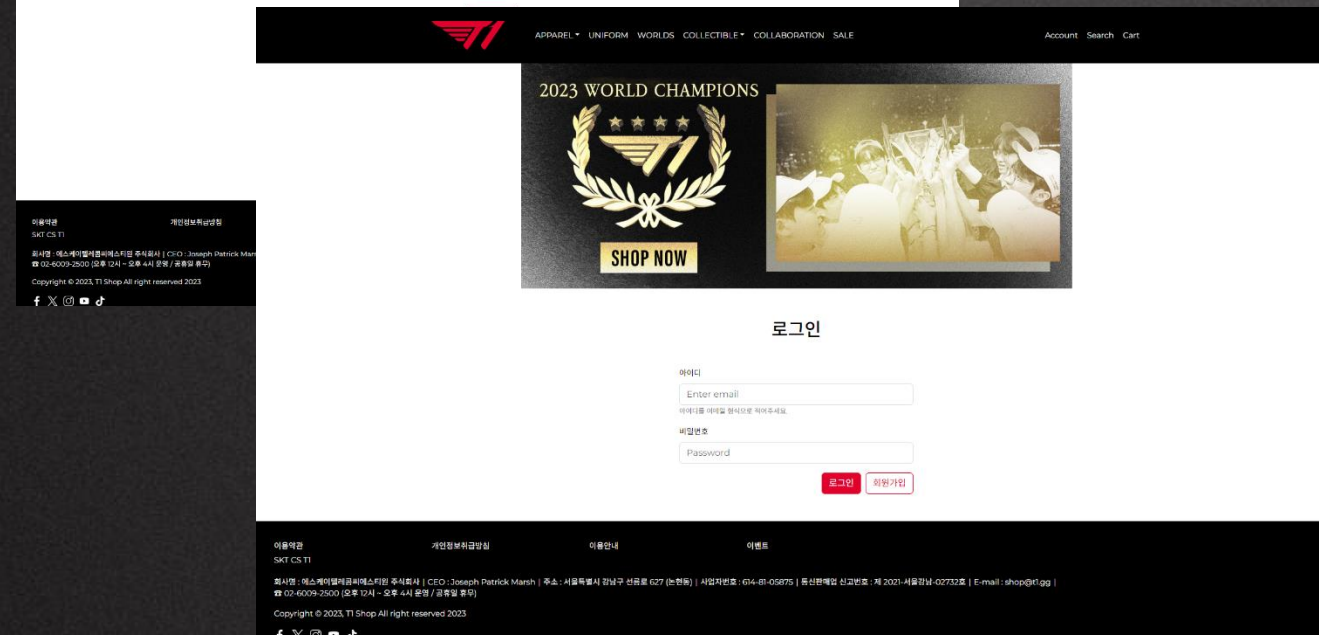
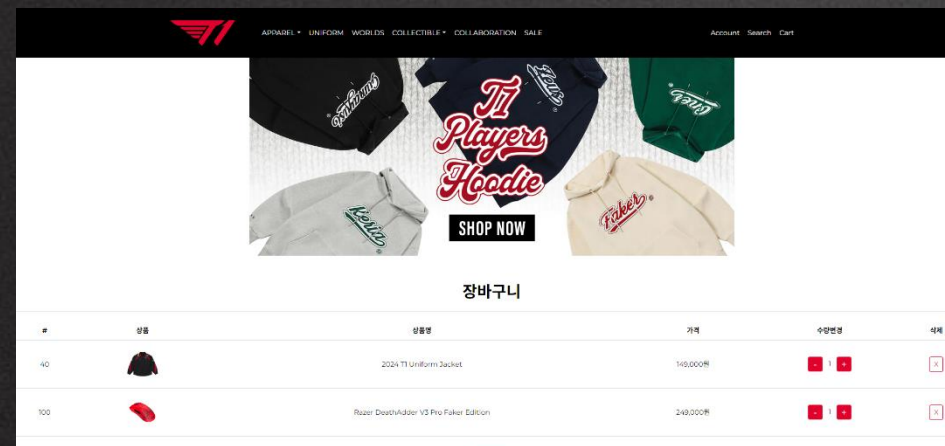
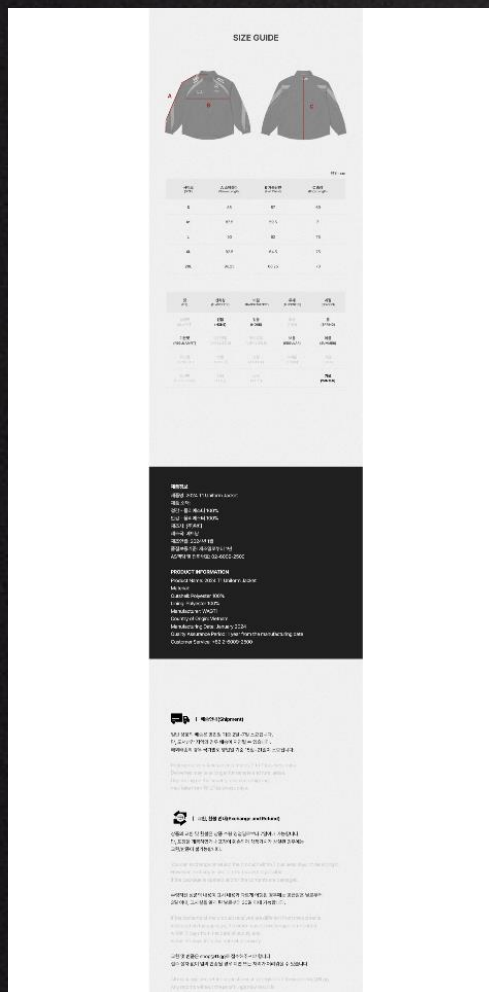
<pc size>

<tab size>

서브페이지 프로토타입

2024

BROTHERHOOD



<detail page>

<cart, login page>

코드 구현화면



2024

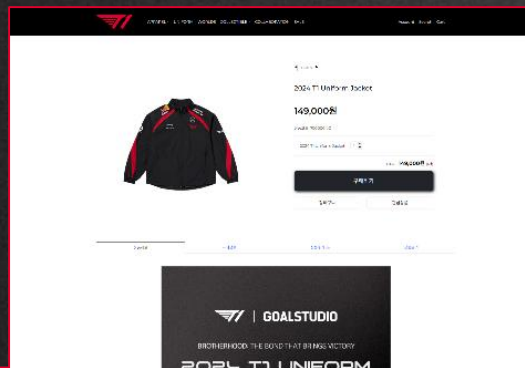
BROTHERHOOD



```
const Newarrival = (props) => {
  const {newItem, no, i} = props;
  let {id} = useParams()
  let selproduct = newItem.find((x)=>x.id==id);
  return (
    <Container className='mb-5'>
      <h2 className='text-center mt-5'>NEW ARRIVAL</h2>
      <p className='bar'></p>
      <p className='text-center mb-5' style={{color:'#aaa'}}>T1의 신상품을 만나보세요</p>
      <Row>
        {
          newItem.map((value,i) =>
            <Product newItem={value} key={i} i={i} no={no} j={2} />
          )
        }
      </Row>
    </Container>
  )
}

const Worlds = (props) => {
  const {newItem, no, i} = props;
  let {id} = useParams()
  let selproduct = newItem.find((x)=>x.id==id);
  return (
    <Container className='mb-5'>
      <h2 className='text-center mt-5'>T1 2023 WORLDS COLLECTION</h2>
      <p className='bar'></p>
      <p className='text-center mb-5' style={{color:'#aaa'}}>2023 WORLDS CHAMPION T1</p>
      <Row>
        {
          newItem.map((value,i) =>
            <Product newItem={value} key={i} i={i} no={no} j={2} />
          )
        }
      </Row>
    </Container>
  )
}

const Uniform = (props) => {
  const {newItem, no, i} = props;
  let {id} = useParams()
  let selproduct = newItem.find((x)=>x.id==id);
  return (
    <Container className='mb-5'>
      <h2 className='text-center mt-5'>T1 OFFICIAL UNIFORM</h2>
      <p className='bar'></p>
      <p className='text-center mb-5' style={{color:'#aaa'}}>T1 유니폼을 입고 T1의 응원해주세요!</p>
      <Row>
        {
          newItem.map((value,i) =>
            <Product newItem={value} key={i} i={i} no={no} j={2} />
          )
        }
      </Row>
    </Container>
  )
}
```



각 제품을 클릭하면 해당 제품의 Detail 페이지가 컴포넌트로 전환되어 SPA로 확인가능하게 구현하였습니다.

해당 제품 데이터는 props로 받아서 비구조화할당을 해준 뒤 사용합니다.

```
const Product = (props) => {
  const {newItem, i, no, j} = props;
  const navigate = useNavigate();
  const navigateTarget = ['/newarrival', '/worlds2023', '/uniforms']
  const likeNumber = [3,1,10,23,0,0,5,10,5,2,3,5,10,2,24,5,7,8,5,6]

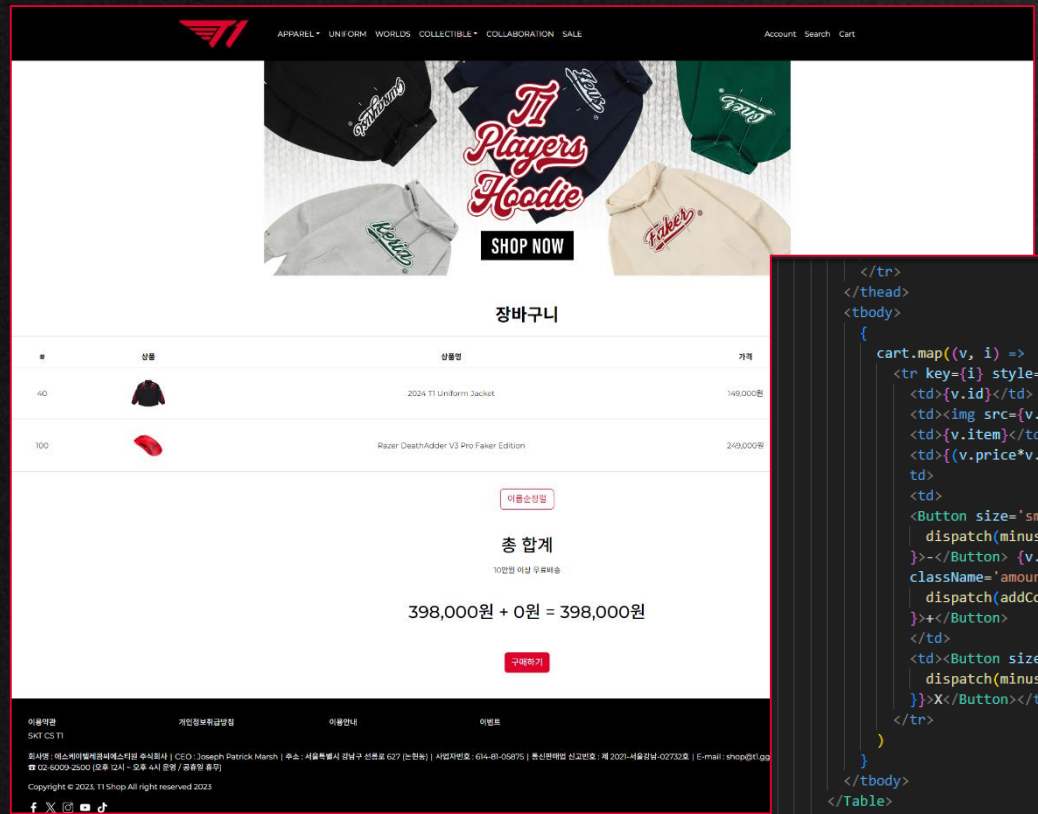
  return (
    <Col md={3} onClick={()=>{navigate(navigateTarget[j]+' /detail/${no[i]}')}}
      className='text-center mb-3 new-item' style={{position:'relative'}}>
      <div className='text-center product-hover-like'>
        <button>LIKE <span>{likeNumber[i]}</span></button>
      </div>
      <img src={process.env.PUBLIC_URL + newItem.imageUrl} width='80%' alt="newItem" />
      <h4 style={{fontSize:'1.1rem', fontWeight:'600'}}>{newItem.title}</h4>
      <hr />
      <p style={{fontSize:'1rem', fontWeight:'700'}}>{newItem.price.toString().replace(/\B(?
        =(\d{3})+(?! \d))/g, ",")}원</p>
      <div className='text-center product-hover-btn'>
        <button><img src={process.env.PUBLIC_URL + "/img/cart.png"} alt="cart" /></button>
        <button><img src={process.env.PUBLIC_URL + "/img/search.png"} alt="search" /></button>
        <button><img src={process.env.PUBLIC_URL + "/img/tabs.png"} alt="tabs" /></button>
      </div>
    </Col>
  )
}
```

React/Components

코드 구현화면

2024

BROTHERHOOD



또 정규표현식을 이용해 가격에 쉼표(,)를 표현해주었습니다.

```
</tr>
</thead>
<tbody>
  {
    cart.map((v, i) =>
      <tr key={i} style={{verticalAlign: 'middle'}}>
        <td>{v.id}</td>
        <td><img src={v.imgUrl} alt="" width={'80px'}></td>
        <td>{v.item}</td>
        <td>{(v.price*v.amount).toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",")}원</td>
        <td>
          <Button size='sm' variant='danger' className='amount-btn me-2' onClick={() =>
            dispatch(minusCount(v.id))
          }>-</Button> {v.amount} <Button size='sm' variant='danger'
            className='amount-btn ms-2' onClick={() =>
            dispatch(addCount(v.id))
          }>+</Button>
        </td>
        <td><Button size='sm' variant='outline-danger' onClick={()=>{
          dispatch(minusItem(v.id))
        }}>X</Button></td>
      </tr>
    )
  }
</tbody>
</Table>
<Button className='mt-3' variant='outline-danger' onClick={()=>{dispatch(sortName())}}>
  이름순정렬</Button>{' '}
<h2 className='mt-5'>총 합계</h2>
<p className='mt-3'>10만원 이상 무료배송</p>
<h2 className='mt-5'> {total().toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",")}원 +
  {shipping.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",")}원 = {(total() + shipping).
  toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",")}원</h2>
<Button className='mt-5 mb-5' variant='danger' onClick={()=>{
  alert('로그인이 필요한 서비스입니다.')
  navigate('/login/')}}>구매하기</Button>
</div>
}
export default Cart
```

```
1 import { configureStore, createSlice } from '@reduxjs/toolkit'
2
3 let cart = createSlice({
4   name: 'cart',
5   initialState: [
6     ],
7   reducers : {
8     addCount(state, action){
9       let idx = state.findIndex((a) => { return a.id === action.payload })
10      state[idx].amount++;
11    },
12
13    minusCount(state, action){
14      let idx = state.findIndex((a) => { return a.id === action.payload })
15
16      if(state[idx].amount <=1){
17        alert('최소 구매수량은 1개 입니다')
18        state[idx].amount=1;
19      } else {
20        --state[idx].amount;
21      }
22    },
23
24    sortName(state){
25      state.sort((a,b) =>
26        a.item > b.item ? 1: -1,
27      );
28    },
29
30    addItem(state, action){
31      let num = state.findIndex(a => a.id === action.payload.id)
32      if (num >= 0){
33        state[num].amount++;
34      } else {
35        state.push(action.payload)
36      }
37    },
38
39    minusItem(state, action){
40      let num = state.findIndex(a => a.id === action.payload)
41      state.splice(num,1)
42    }
43  }
44 }
```

장바구니 페이지는 Redux를 이용해
상품의 정보를 받거나 개수를 버튼으로
늘리거나 줄이고, 버튼을 클릭해 이름순으로
정렬하고 삭제 버튼으로 해당 상품을 장바구니
에서 삭제하도록 구현했습니다.

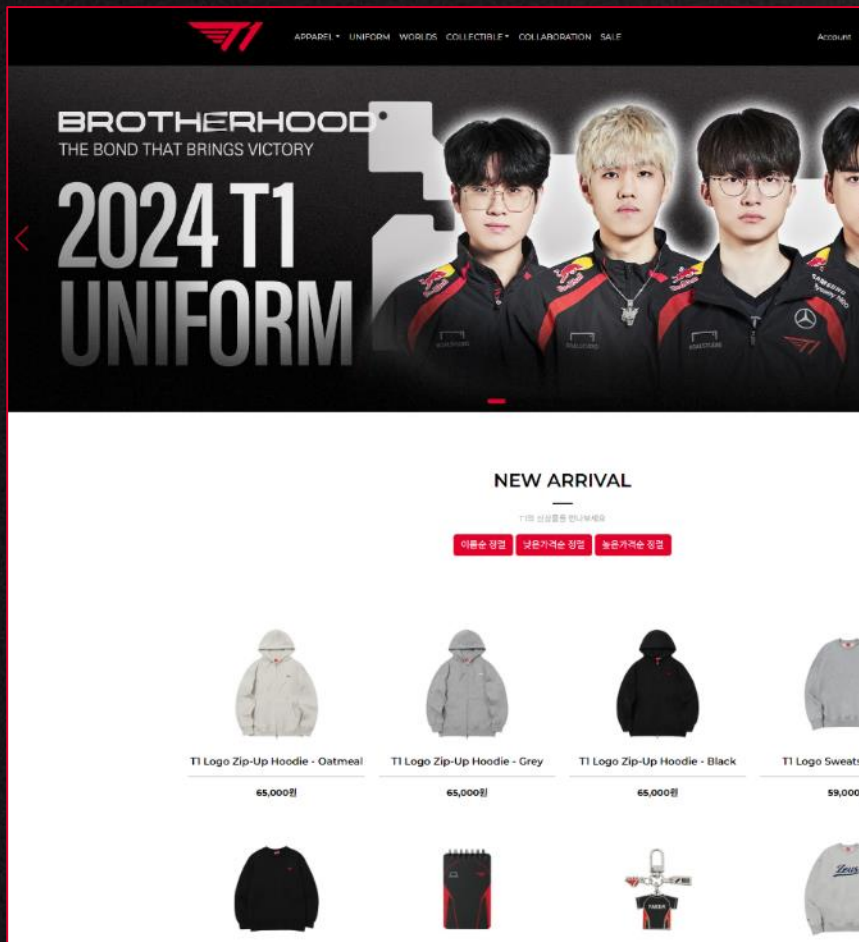
React/Cart.jsx

코드 구현화면



2024

BROTHERHOOD



```
const Home = (props) => {
  const {newItem, setNewItem, no, setNo} = props;
  const [sale, setSale] = useState(data4);
  const [count, setCount] = useState(1);
  const [isHover, setIsHover] = useState(false);
  const toggleHover = () => setIsHover(!isHover);
  const navigate = useNavigate();

  const sortTitle = () => {
    let copy = [...newItem].sort((a,b)=>(a.title > b.title) ? 1:-1)
    setNewItem(copy)
    let no_copy=[];
    for (let i in copy){
      no_copy.push(copy[i].id)
    }
    setNo(no_copy)
  }

  const incPrice = ()=> {
    let copy = [...newItem].sort((a,b)=>(a.price > b.price) ? 1:-1)
    setNewItem(copy)
    let no_copy=[];
    for (let i in copy){
      no_copy.push(copy[i].id)
    }
    console.log(no_copy)
    setNo(no_copy)
  }

  const decPrice = ()=> {
    let copy = [...newItem].sort((a,b)=>(a.price < b.price) ? 1:-1)
    setNewItem(copy)
    let no_copy=[];
    for (let i in copy){
      no_copy.push(copy[i].id)
    }
    setNo(no_copy)
  }
}
```

메인페이지 역할인 Home.jsx의 경우 최상단의 'Newarrival' 항목에 정렬 버튼을 3개 구현하여, 클릭에 따라 제품의 정렬이 변화하도록 구현하였습니다.

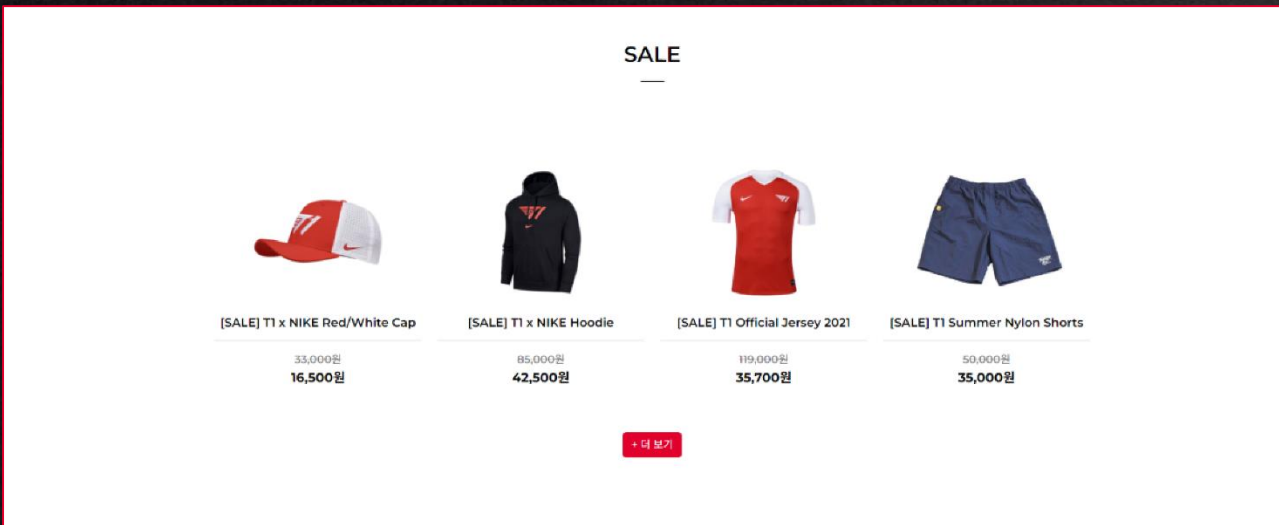
또 useParams를 이용하여 정렬이 변화하더라도 클릭한 상품과 Detail이 어긋나지 않도록 id를 맞춰주었습니다.

React/Home.jsx

코드 구현화면

2024

BROTHERHOOD



‘Home.jsx’의 하단 Sale 항목에는 Git에 업로드 한 json 파일 데이터를 axios 를 이용하여 불러와서 화면에 표시하는 코드를 구현하였습니다.

```
<Container>
  <Row>
    {sale.map((value, i) =>
      <Sale sale={value} key={i} no={no} i={i} j={3} />)
    }
  </Row>
</Container>

<div style={{textAlign:'center',margin:'50px auto'}}>
  <Button variant="danger" onClick={()=>{
    if (count===1) {
      axios.get('https://alanpark727.github.io/react_data/sale2.json').then(result => {
        let copy = [...sale, ...result.data];
        setSale(copy);
        setCount(2);
      })
    } else if (count===2) {
      axios.get('https://alanpark727.github.io/react_data/sale3.json').then(result => {
        let copy = [...sale, ...result.data];
        setSale(copy);
        setCount(3);
      })
    } else if (count===3) {
      axios.get('https://alanpark727.github.io/react_data/sale4.json').then(result => {
        let copy = [...sale, ...result.data];
        setSale(copy);
        setCount(4);
      })
    } else if (count===4) {
      axios.get('https://alanpark727.github.io/react_data/sale5.json').then(result => {
        let copy = [...sale, ...result.data];
        setSale(copy);
        setCount(5);
      })
    } else {
      alert('더 이상 상품이 없습니다.');
```

React/Home.jsx


```
const Product = (props) => {  
  const {newItem, i, no, j} = props;  
  const navigate = useNavigate();  
  const navigateTarget = ['/newarrival', '/worlds2023', '/uniforms']  
  const likeNumber = [3,1,10,23,0,0,5,10,5,2,3,5,10,2,24,5,7,8,5,6]  
  
  return (  
    <Col md={3} onClick={()=>{navigate(navigateTarget[j]+' /detail/${no[i]}')}}  
      className='text-center mb-3 new-item' style={{position:'relative'}}>  
      <div className='text-center product-hover-like'>  
        <button>LIKE <span>{likeNumber[i]}</span></button>  
      </div>  
      <img src={process.env.PUBLIC_URL + newItem.imageUrl} width='80%' alt="newItem" />  
      <h4 style={{fontSize:'1.1rem', fontWeight:'600'}}>{newItem.title}</h4>  
    </Col>  
  )  
}
```

‘App.jsx’에서 각 제품 정보와 Detail 컴포넌트로 state를 내려줄 때 기본적인 틀을 공유하기 위해서 전달해주는 props를 바꿔줌으로써 각 컴포넌트의 재사용성을 높였습니다.

```
<Routes>  
  <Route path="/" element={<Home newItem={newItem} setNewItem={setNewItem} no={no} setNo={setNo} />} />  
  <Route path="*" element={<div><h2>404</h2>Page Not Found</div>} />  
  <Route path="/newarrival/" element={<div><Newarrival newItem={newItem} setNewItem={setNewItem} no={no} setNo={setNo}/></div>} />  
  <Route path="/newarrival/detail/:id" element={<Detail newItem={newItem} />} />  
  <Route path="/worlds2023/" element={<div><Worlds newItem={worlds} setNewItem={setWorlds} no={no2} setNo={setNo2} /></div>} />  
  <Route path="/worlds2023/detail/:id" element={<Detail newItem={worlds} />} />  
  <Route path="/uniforms/" element={<div><Uniform newItem={uniform} setNewItem={setUniform} no={no3} setNo={setNo3} /></div>} />  
  <Route path="/uniforms/detail/:id" element={<Detail newItem={uniform} />} />  
  <Route path="/login" element={<div><Login /></div>} />  
  <Route path="/cart" element={<div><Cart /></div>} />  
  <Route path="/collaboration/razer" element={<div><Razor /></div>} />  
</Routes>
```


이번 프로젝트는 ‘React’를 이용하여 웹페이지를 구현하는 것이었습니다. 새로운 것을 배우는 것은 언제나 설레고 두근거리는 일입니다. 기존의 html이나 자바스크립트, 제이쿼리랑은 페이지를 만드는 방법이 달라서 신선했습니다. 컴포넌트를 이용해 조각조각 구성품을 만들어서 웹페이지를 조립하는 방식은 마치 프라모델을 조립하는 기분이 들게 했습니다.

또 이번 프로젝트를 진행하면서 ‘bootstrap’도 사용하게 되었는데, 틀이 짜여있는 라이브러리를 이용해 컴포넌트를 구성하고 웹페이지를 만드는 것이 얼마나 수월한 지를 잘 알 수 있었습니다. 특히 가장 크게 느껴졌던 부분은 반응형 옵션들로, 직접 창 크기별로 css를 나누는 과정이 생략되어 빠르게 제작할 수 있었습니다. 물론 기존의 방식과는 다르게 새로운 방법이었기에 프로젝트를 진행하는 것에 있어 시행착오는 겪었지만, 이렇게 새로운 경험을 통해 웹페이지 개발에 대해 조금 더 알게 된 거 같아 만족스럽습니다.

프로젝트를 진행하는 동안 수업에서 배운 내용은 충실하게 반영하고, 프로젝트 진행 기간과 제 개인적인 역량을 잘 가늠하여 정해진 시간 안에 만족할만한 퀄리티의 웹페이지를 구성한다는 작은 목표를 세웠습니다. 지나치게 욕심을 부려서 진행기간이 너무 길어지거나 하지 않도록 기능 구현에 필요한 부분과 조금은 욕심인 부분을 줄이는 방향으로 진행했습니다. 결과적으로 개인 컨디션 문제도 겹치고 해서 예상보다 완성에 시간이 좀 더 걸리긴 했지만, 배운 기능은 착실하게 쓴다, 라는 대전제는 지킬 수 있어서 다행이라고 생각합니다.

앞으로도 더 많은 공부를 통해 ‘React’ 뿐 아니라 다양한 프레임워크들을 접하고 다룰 수 있게 되도록 개인적으로도 공부를 계속 해야겠다는 생각이 들었습니다.

