

A decorative graphic on the left side of the slide, consisting of a large white circle and a smaller white circle, both centered on the left edge. The circles are partially cut off by the left edge of the slide.

React Project

강누리

목차

진행기간: 2023.10.4 ~ 10.25



Theme

✓ 카테고리

의류와 관련된 페이지를 제작해 보고 싶었는데 FILA의 홈페이지는 컨셉이 있는 운동복과 신발, 테니스 커뮤니티 등 재밌고 다양한 주제들이 있어 선택하게 되었습니다.

✓ 스타일

오토바이를 활용한 Motorcore 컨셉이 옷과 모델의 분위기에 잘 어울렸고, 전체적으로 흰색과 네이비색이 대비되어 깔끔한 분위기가 눈에 들어왔습니다.

선정 계기

Layout

✗ 구조

메인페이지의 첫 섹션이 화면을 꽉 채우는 이미지 슬라이드여서 옷의 컨셉을 더 잘 느낄 수 있었고, 상세페이지에서 스크롤을 활용한 이벤트들이 있어서 프로젝트에 적용해 보면 좋겠다는 생각이 들었습니다.

✗ 슬라이드

비슷한 구조이지만 이미지의 크기가 달라져 각 섹션의 주제를 강조하고, 구분하기 쉽게 만들어져 선택하게 되었습니다.

Design

편안한 느낌을 주면서도 배색을 활용한 디자인으로 스포티하게 보일 수 있는 라이더의 스타일을 표현한 컨셉입니다.



컨셉

Motorcore

페르소나

이해린

26세

성격: 활발함

취미: 활동적인 것을 좋아해 운동을 좋아하고 쉬는날엔 주로 외출하는 편이다.



동기

운동으로는 주로 공원에서 런닝을 하고 가끔 친구들과 배드민턴을 치기도 한다.

보통 평일 저녁에 운동을 하는편이라 반팔을 입고 나갔는데 최근엔 땀을 흘리고 나면 쌀쌀한 느낌이 들어 운동할 때 가볍게 걸칠 수 있는 겹옷을 사려고 한다.

목적 & 니즈

운동을 하며 들고다니기 편한 가벼운 자켓을 찾고 있다. 예쁜 운동복을 입으면 기분 좋게 운동할 수 있어서 디자인도 신경쓴 운동복을 좋아한다.

페르소나

박재현

30세

성격: 외향적이고 주변사람들에게 친절하다.

취미: 음악 듣기, 게임



동기

옷입는 것을 좋아해 다양한 종류의 옷을 가지고 있다.

회사에서는 주로 깔끔하고 단정한 셔츠나 슬랙스, 코트 등을 입는데 주말에는 편하게 입는것을 좋아한다.

주말에 나갈일이 많지 않다보니 일할 때 입을 옷들만 사다가 최근 주말에 친구와 만날일이 많아져 편하게 입을 트레이닝복을 둘러보는 중이다.

목적 & 니즈

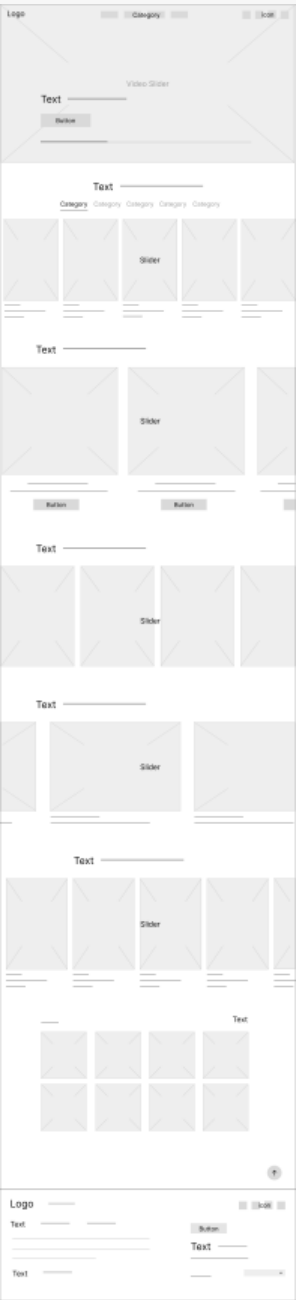
편하게 입을 수 있는 옷이지만 꾸안꾸 느낌을 낼 수 있는 옷을 찾고 있다.

코디하기 쉽고 활용도가 높은 셋업을 위주로 보고있다.

와이어프레임 비디오 슬라이드와 이미지 슬라이드로 구성했습니다.

프로토타입 섹션별로 슬라이드의 이미지 크기가 달라집니다.

PC



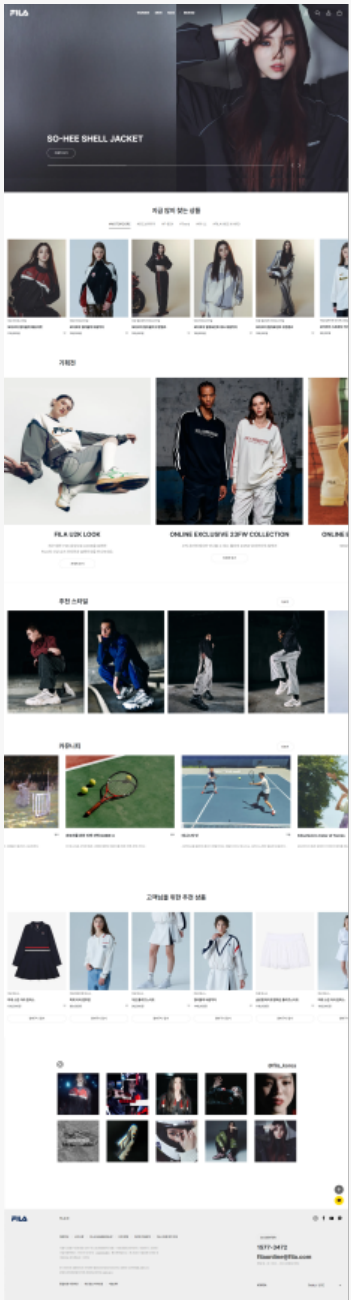
Tablet



Mobile



PC



Tablet



Mobile



와이어프레임

스크롤을 내리면 상단의 텍스트 박스가 따라 내려옵니다.

프로토타입

옷의 상세 이미지와 컨셉을 나타내는 사진으로 구성했습니다.

PC

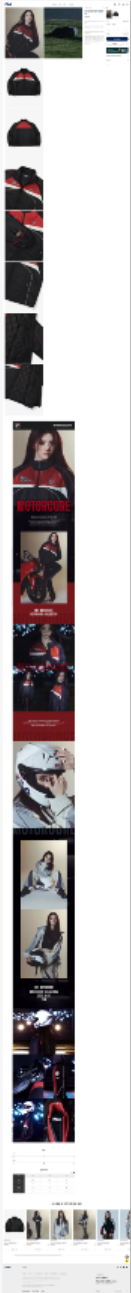
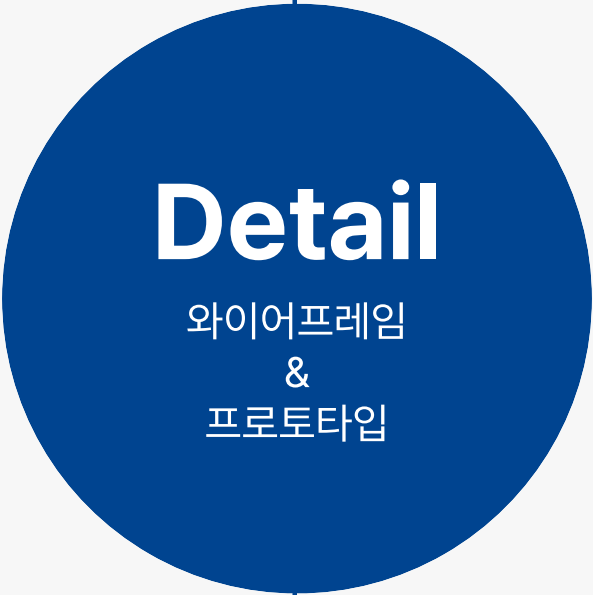
Tablet

Mobile

PC

Tablet

Mobile



와이어프레임

스크롤을 내리면 상단의 텍스트 박스가 따라 내려옵니다.

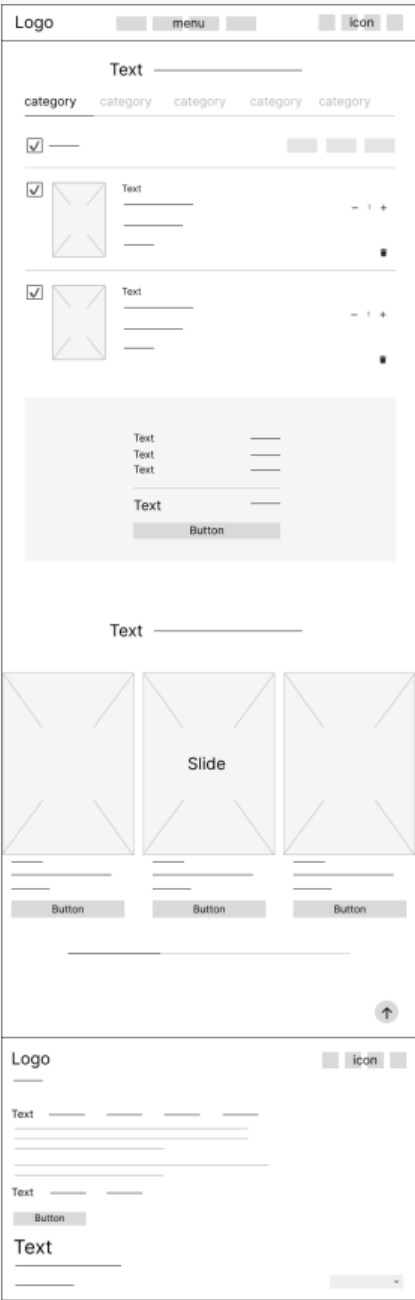
프로토타입

옷의 상세 이미지와 컨셉을 나타내는 사진으로 구성했습니다.

PC

Tablet

Mobile

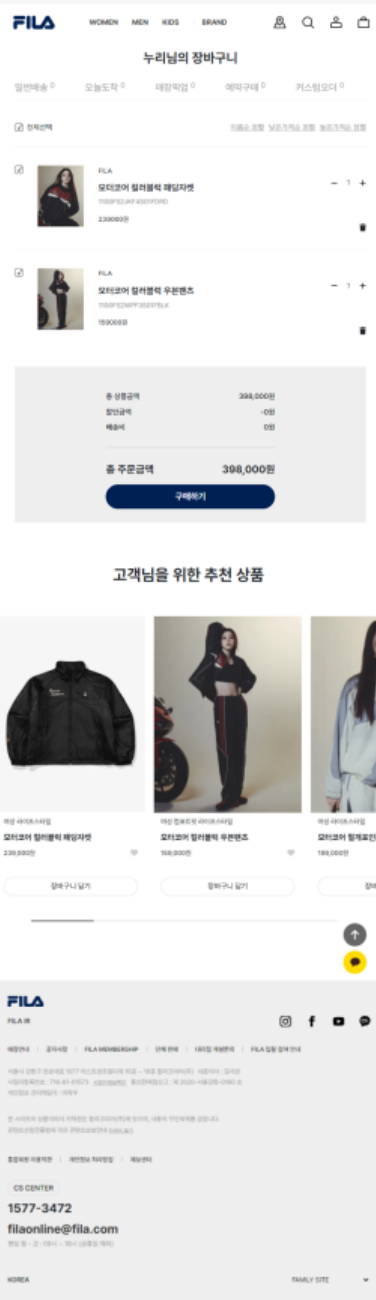
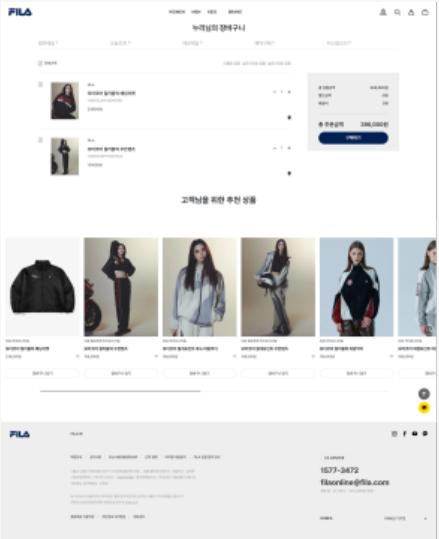


Cart
와이어프레임
&
프로토타입

PC

Tablet

Mobile



FILA' s
스타일



Color

#002053, #fff



Font

힐라 FILA
힐라 FILA
힐라 **FILA**



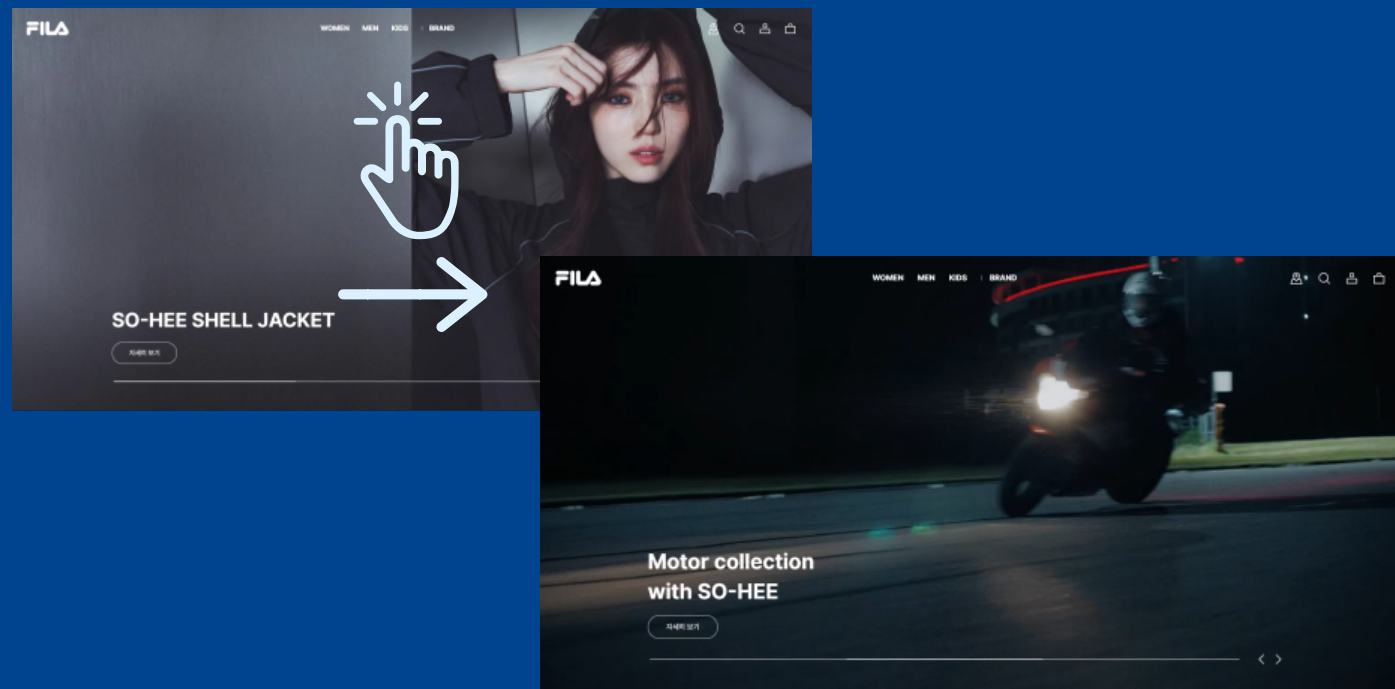
Tennis

item
&
Community

Video Slide

화면을 꽉채우는 이미지와 비디오 .

손으로 옆으로 넘길 수 있고, 화살표 버튼으로도 이동이 가능합니다.



SCSS

```
.swiper-slide { position: relative;
  display: flex; flex-direction: column; align-items: center; justify-content: center;

  img { display: block; width: 100%; }

  video { display: block; width: 100%; }

  .swiper_txt { width: 75%; position: absolute; top: 73%;

    h1 { color: #fff; font-weight: 700; font-size: 3rem; }
    .swiper_btn { border-radius: 30px; padding: 13px 40px; }
    &.swiper_txt2 { top: 66%; }
  }
}

.swiper-scrollbar.swiper-scrollbar-horizontal {
  background: #aaa; height: 0.12rem; width: 68.5%; bottom: 9%; left: 12.7%;

  .swiper-scrollbar-drag { background: #fff; }
}

.swiper-button-prev {
  top: 91%; left: 83%; color: #fff;

  &::after { display: inline-block; font-size: 1.3rem; }
}
```

Swiper

```
(/* main banner */)
<Swiper
  slidesPerView={1}
  grabCursor={true}
  keyboard={{
    enabled: true,
  }}
  scrollbar={true}
  navigation={true}
  modules={[Keyboard, Scrollbar, Navigation, Autoplay]}
  autoplay={{
    delay: 14000,
    disableOnInteraction: false,
  }}
  loop={true}
  className="mySwiper mainSwiper"
>
  <SwiperSlide>
    <img src={process.env.PUBLIC_URL + "/img/banner1.jpg"} />

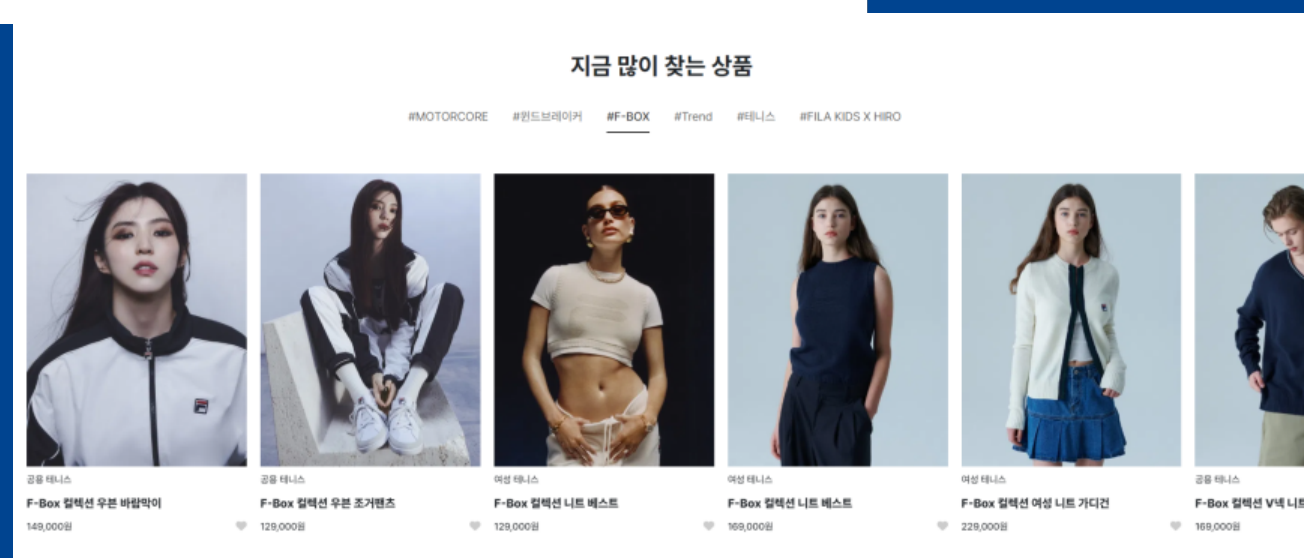
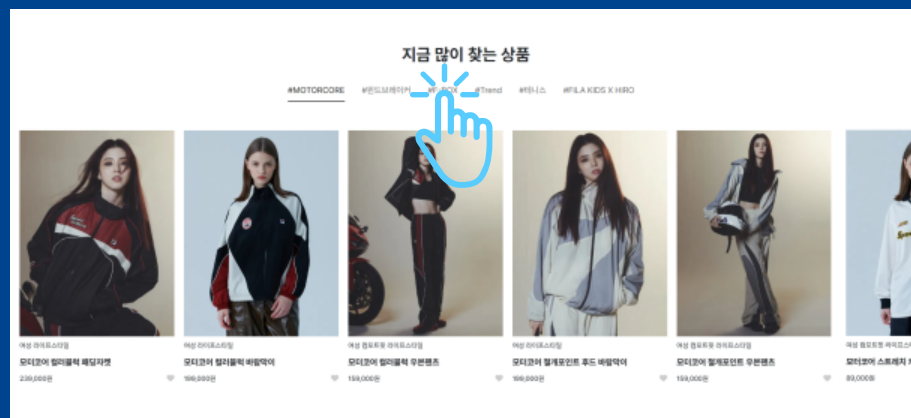
    <div className="swiper_txt">
      <h1>SO-HEE SHELL JACKET</h1>
      <Button variant="outline-light" className='swiper_btn mt-3'>자세히 보기</Button>
    </div>
  </SwiperSlide>
  <SwiperSlide>
    <video src={process.env.PUBLIC_URL + "/img/banner2.mp4"} autoPlay loop muted</video>

    <div className="swiper_txt swiper_txt2">
      <h1>Motor collection</h1>
      <h1>with SO-HEE</h1>
      <Button variant="outline-light" className='swiper_btn mt-3'>자세히 보기</Button>
    </div>
  </SwiperSlide>
  <SwiperSlide>
    <video src={process.env.PUBLIC_URL + "/img/banner3.mp4"} autoPlay loop muted</video>

    <div className="swiper_txt">
      <h1>HAILEY FOR FILA</h1>
      <Button variant="outline-light" className='swiper_btn mt-3'>자세히 보기</Button>
    </div>
  </SwiperSlide>
</Swiper>
```

Navs and tabs

Redux로 받아온 변수를 이용해 각 메뉴의 index를 정해주고, 하위 컴포넌트에서 Swiper 배열에 index를 연결해 원하는 메뉴를 탭하면 각각의 탭에 맞는 상품이 보여지게 합니다.



Redux

```
let tab = createSlice({
  name: 'tab',
  initialState: {value: 0},
  reducers:{
    chTab(state, action){
      state.value = action.payload
    }
  }
})

const NowTab = (props) => {
  const navigate = useNavigate();
  let tab = useSelector(state => state.tab)
  let { moterCoreData } = props;
```

Bootstrap

```
/* section1 */
<div className='now'>
  <h1>지금 많이 찾는 상품</h1>

  <Nav variant='underline' defaultActiveKey='motorCore' className='mt-4'>
    <Nav.Item>
      <Nav.Link eventKey='motorCore' onClick={() => { dispatch(chTab(0)) }}>#MOTORCORE</Nav.Link>
    </Nav.Item>
    <Nav.Item>
      <Nav.Link eventKey='wind' onClick={() => { dispatch(chTab(1)) }}>#윈드브레이커</Nav.Link>
    </Nav.Item>
    <Nav.Item>
      <Nav.Link eventKey='fBox' onClick={() => { dispatch(chTab(2)) }}>#F-BOX</Nav.Link>
    </Nav.Item>
    <Nav.Item>
      <Nav.Link eventKey='trend' onClick={() => { dispatch(chTab(3)) }}>#Trend</Nav.Link>
    </Nav.Item>
    <Nav.Item>
      <Nav.Link eventKey='tennis' onClick={() => { dispatch(chTab(4)) }}>#테니스</Nav.Link>
    </Nav.Item>
    <Nav.Item>
      <Nav.Link eventKey='kids' onClick={() => { dispatch(chTab(5)) }}>#FILA KIDS X HIRO</Nav.Link>
    </Nav.Item>
  </Nav>

  <NowTab moterCoreData={moterCoreData} />
</div>
```

Swiper

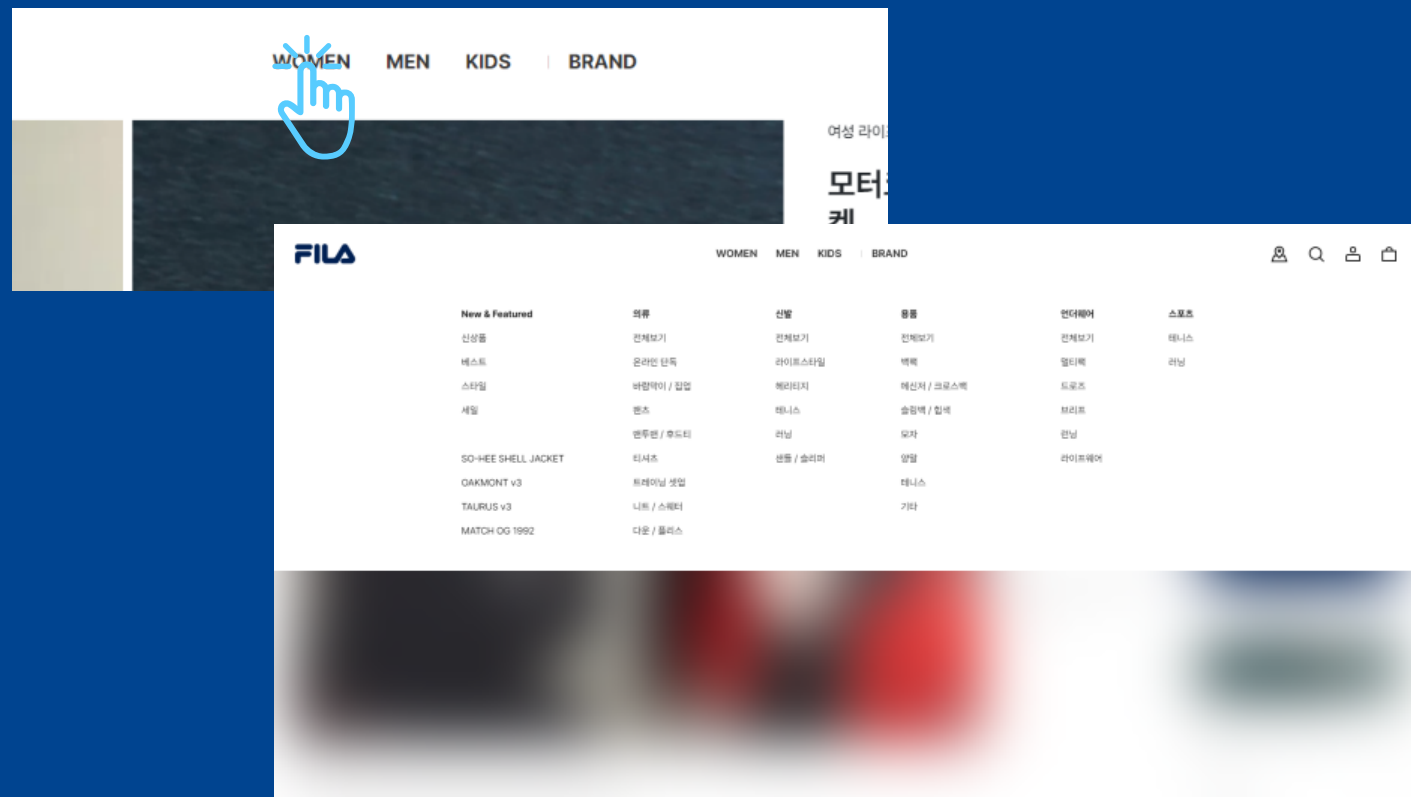
```
{
  moterCoreData.map((v, i) =>
    <SwiperSlide key={i}>
      <NavLink onClick={()=>{navigate(`/detail/${i}`)}} >
        <img src={process.env.PUBLIC_URL + v.imgUrl} alt="shell jacket"/>
      </NavLink>

      <div className='now_txt'>
        <h6>{v.type}</h6>
        <NavLink onClick={()=>{navigate(`/detail/${i}`)}}><h5>{v.name}</h5></NavLink>
        <div style={{ display: 'flex', justifyContent: 'space-between' }}>
          <h6>{v.price.toLocaleString()}원</h6>
          <FontAwesomeIcon icon={faHeart} style={{ color: '#ccc' }} />
        </div>
      </div>
    </SwiperSlide>
  ),
  kidsData.map((v, i) =>
    <SwiperSlide key={i}>
      <NavLink onClick={()=>{navigate(`/detail/${i}`)}}>
        <img src={process.env.PUBLIC_URL + v.imgUrl} alt="shell jacket" />
      </NavLink>

      <div className='now_txt'>
        <h6>{v.type}</h6>
        <NavLink onClick={()=>{navigate(`/detail/${i}`)}}><h5>{v.name}</h5></NavLink>
        <div style={{ display: 'flex', justifyContent: 'space-between' }}>
          <h6>{v.price}</h6>
          <FontAwesomeIcon icon={faHeart} style={{ color: '#ccc' }} />
        </div>
      </div>
    </SwiperSlide>
  )][tab.value]
}
</Swiper>
```

Navigation

메뉴에 Hover하면 숨겨져 있던 카테고리 박스가 나옵니다.



React

```
let [showUnderNav, setShowUnderNav] = useState({ display: 'none' });
let [categories, setCategories] = useState({ color: 'ffff' });
let [navPosition, setNavPosition] = useState({ position: 'absolute' });

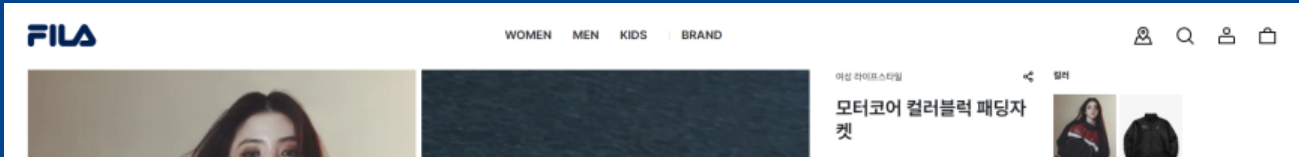
function navMouseOn(e) {
  setShowUnderNav({ display: 'block' });
  setCategories({ color: '#333' });

  nav.current.style.background = 'ffff';
  logo.current.style.backgroundImage = 'url("data:image/svg+xml,X3Csvg xmlns="http://www.w3.org/2000/svg" width="100" height="33" e.target.style.borderBottom = "1px solid #333";
  location.current.innerHTML = '
  <Nav.Link onClick={()=>{navigate("/product")}} className='category' style={categories} onMouseOver={(e) => {
    if (window.matchMedia("(min-width:768px)").matches) {
      navMouseOn(e);
    }
  }} onMouseOut={outNav}>WOMEN</Nav.Link>
  <Nav.Link onClick={()=>{navigate("/product")}} className='category' style={categories} onMouseOver={(e) => {
    if (window.matchMedia("(min-width:768px)").matches) {
      navMouseOn(e);
    }
  }} onMouseOut={outNav}>MEN</Nav.Link>
  <Nav.Link onClick={()=>{navigate("/product")}} className='category' style={categories} onMouseOver={(e) => {
    if (window.matchMedia("(min-width:768px)").matches) {
      navMouseOn(e);
    }
  }} onMouseOut={outNav}>KIDS</Nav.Link>
  <Nav.Link className='category' style={categories}></Nav.Link>
  <Nav.Link onClick={()=>{navigate("/product")}} className='category' style={categories} onMouseOver={(e) => {
    if (window.matchMedia("(min-width:768px)").matches) {
      navMouseOn(e);
    }
  }} onMouseOut={outNav}>BRAND</Nav.Link>
</div>

<div className='opacity_block' onMouseOver={() => {
  if (location.pathname === '/') {
    navMouseOut()
  } else {
    setShowUnderNav({ display: 'none' });
  }
}}></div>
```

Navigation

메인페이지 외에 다른 페이지들의 메뉴, 로고, 아이콘의 색상을 다르게 적용했습니다.



React

```
useEffect(() => {
  if (location.pathname !== '/') {
    setNavPosition({position: 'fixed'});
    setCategories({ color: '#333' });

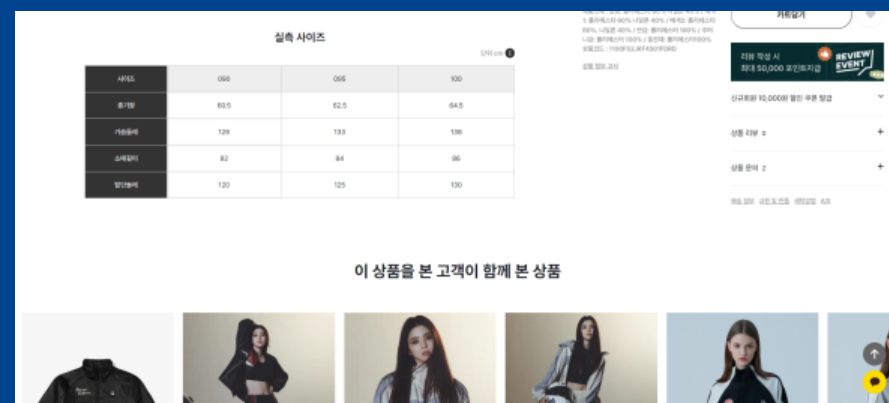
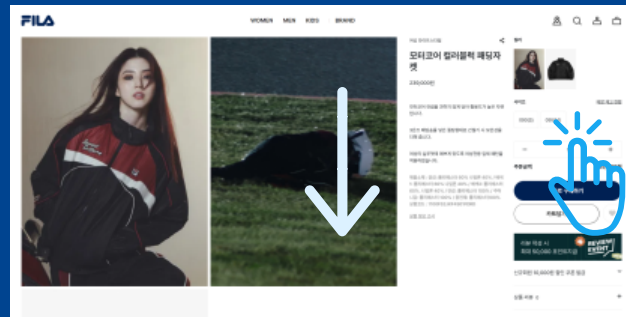
    nav.current.style.background = '#fff';
    logo.current.style.backgroundImage = `url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' width='100' height='26.852' viewBox='0 0 100 26.852'%3E%3Crect width='100' height='26.852' fill='white'/%3E%3C/svg%3E")`;
    location.current.innerHTML = ``;
    search.current.innerHTML = ``;
    login.current.innerHTML = ``;
    basket.current.innerHTML = ``;
  } else {
    setNavPosition({position: 'absolute'});
    setShowUnderNav({ display: 'none' });
    setCategories({ color: '#fff' });

    nav.current.style.background = 'transparent';
    logo.current.style.backgroundImage = `url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' width='100' height='26.852' viewBox='0 0 100 26.852'%3E%3Crect width='100' height='26.852' fill='white'/%3E%3C/svg%3E")`;
    location.current.innerHTML = ``;
    search.current.innerHTML = ``;
    login.current.innerHTML = ``;
    basket.current.innerHTML = ``;
  }
}, [location.pathname])
```

Scroll

스크롤을 내리면 비디오와 텍스트 박스가 따라 내려오다가 자신의 섹션을 벗어나면 고정됩니다.

텍스트 박스는 자신의 위치에서 스크롤 하면 따로 스크롤 됩니다.



SCSS - position

```
.detail_right{position: sticky; top: 100px; display: flex; height: 90vh; padding-right: 0 !important;

.d_r_left{width: calc((100% + 40px) / 2 - 40px); padding-left: 30px;

  &>div{display: flex; justify-content: space-between; flex-wrap: wrap;}
  span{ font-size: 0.85rem; font-weight: 500; }
  h3{margin: 20px 0 20px 0; font-weight: 700;}
  p{font-weight: 500;
    &:not(:first-of-type){font-size: 0.87rem;}
    &:first-of-type{margin-bottom: 50px;}
    &:nth-of-type(2),&:nth-of-type(3),&:nth-of-type(4){margin-bottom: 30px;}
    &:nth-of-type(5){margin-bottom: 0px;}
    &:last-of-type{text-decoration: underline; cursor: pointer;}
    &:nth-of-type(5),&:nth-of-type(6),&:last-of-type{color: #999; font-weight: 600; font-size: 0.85rem;}
  }

.d_l_t_right{ width: calc(50% - 4px); height: 39.55vw; top: 100px; position: sticky; overflow: hidden;
  margin-bottom: 8px;
  & > video{width: 100%; padding: 0 0 8px 4px; position: absolute; top: 50%; left: 50%; transform: translate(-50%,-50%); }
```

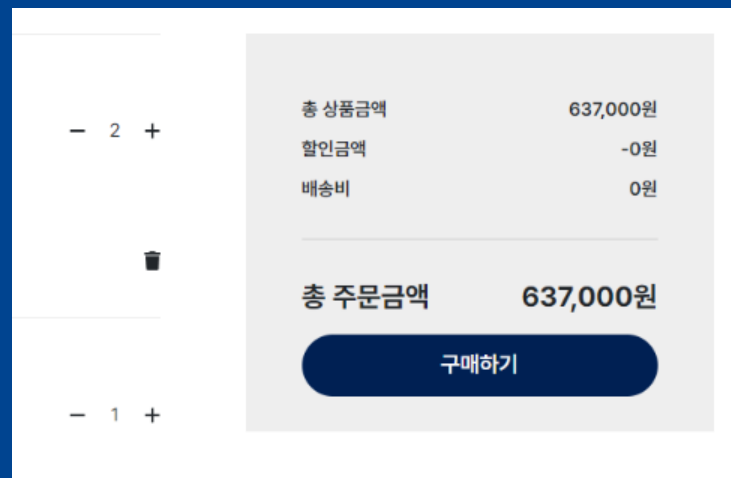
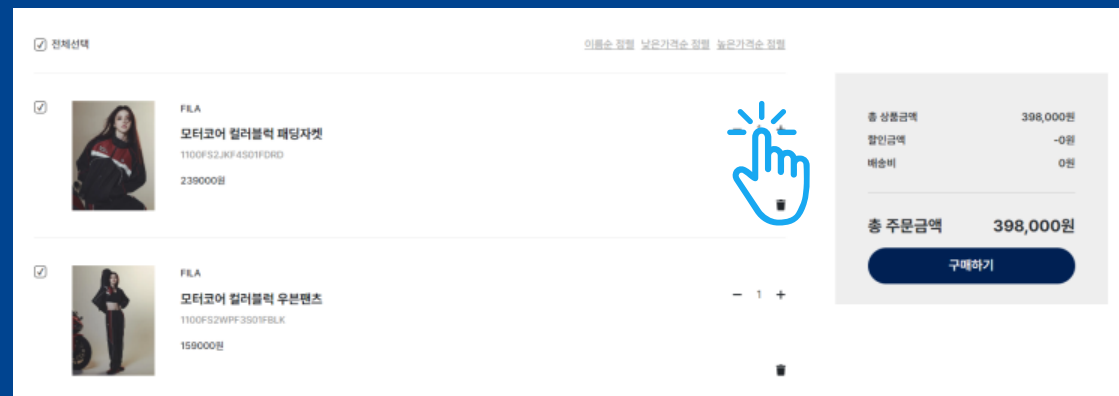
SCSS - overflow

```
.d_r_right{width: calc((100% + 40px) / 2 - 40px); height: 100%; padding-left: 30px; flex-basis: 53%;
overflow-y: scroll;
&::-webkit-scrollbar{display: none;}

h6{font-weight: 800; font-size: 0.85rem; margin-bottom: 20px;}
&>img:first-of-type, &>img:nth-of-type(2){width: 28%; margin-right: 5px; min-width: 75px;}
&>div{display: flex; justify-content: space-between;
  &:not(:first-of-type){ margin-top: 25px; flex-wrap: wrap;}
  &:last-of-type{ display: block; margin: 20px 0;
    span{font-size: 0.85rem; color: #888; text-decoration: underline; margin-right: 10px;}
  }
  p{font-weight: 800; }
```

Count

상품의 수량을 추가하고 뺄때마다 상품금액, 배송비, 총 주문금액이 달라집니다.



Redux

```
reducers:{
  addCount(state, action){
    let num = state.findIndex(a => a.id == action.payload)
    state[num].count++
  },

  minusCount(state, action){
    let num = state.findIndex(a => a.id == action.payload)
    state[num].count--

    if(state[num].count < 1){
      alert('최소 주문수량은 1개입니다.')
      state[num].count = 1
    }
  },
}
```

React

```
let tab = useSelector(state => state.tab.value)
let cart = useSelector(state => state.cart)
let dispatch = useDispatch()

let total = ()=>{
  let sum = 0;
  for(let i in cart){
    sum += cart[i].price * cart[i].count
  }
  return sum
}

let [deliveryFee, setDeliveryFee] = useState(2500)
useEffect(()=>{
  function fee() {
    if (total() >= 50000) {
      setDeliveryFee(0)
    } else {
      setDeliveryFee(2500)
    }
  }
  fee()
})
```

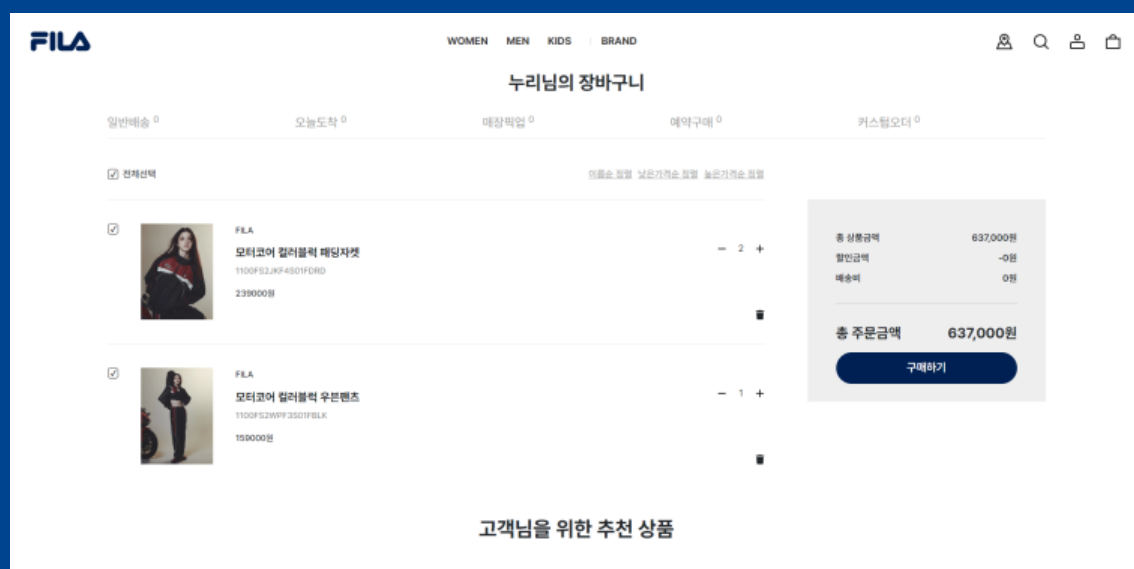
```
<FontAwesomeIcon icon={faMinus} className='faPlus' style={{cursor:'pointer'}}
  onClick={()=>{dispatch(minusCount(v.id))}}
/>
<span>{v.count}</span>
<FontAwesomeIcon icon={faPlus} className='faMinus' style={{cursor:'pointer'}}
  onClick={()=>{dispatch(addCount(v.id))}}
/>
```

```
<div className='cart_priceBox p-5'>
  <div>
    <span>총 상품금액</span>
    <span>{total().toLocaleString()}원</span>
  </div>
  <div>
    <span>할인금액</span>
    <span>-0원</span>
  </div>
  <div>
    <span>배송비</span>
    <span>{deliveryFee.toLocaleString()}원</span>
  </div>

  <div>
    <h4>총 주문금액</h4>
    <h4>{(total() + deliveryFee).toLocaleString()}원</h4>
    <button>구매하기</button>
  </div>
</div>
```

Add item

상세 페이지에서 카트담기를 누르면 장바구니 페이지에 상품이 추가됩니다.
이미 있는 상품은 수량만 증가됩니다.



Redux

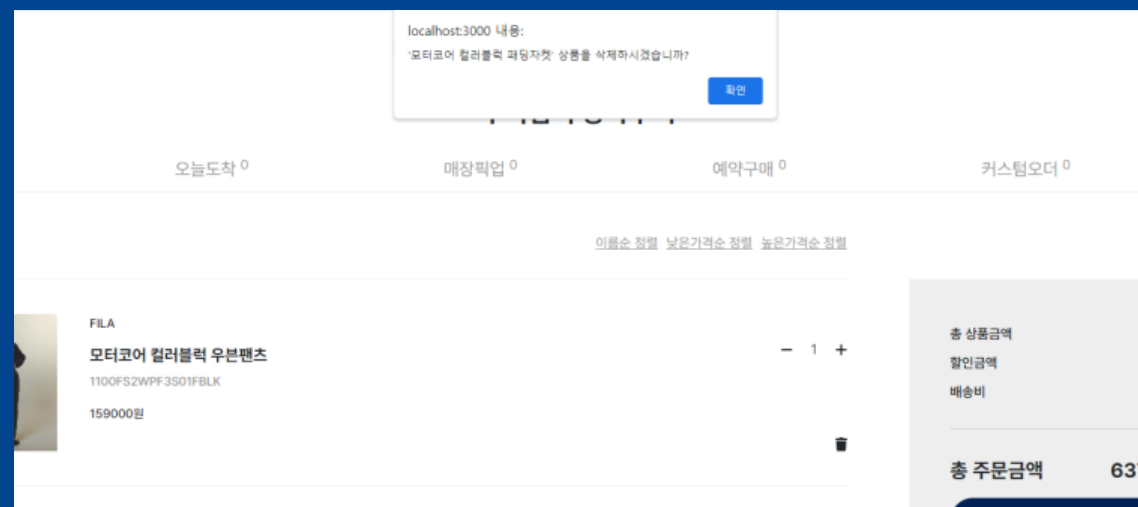
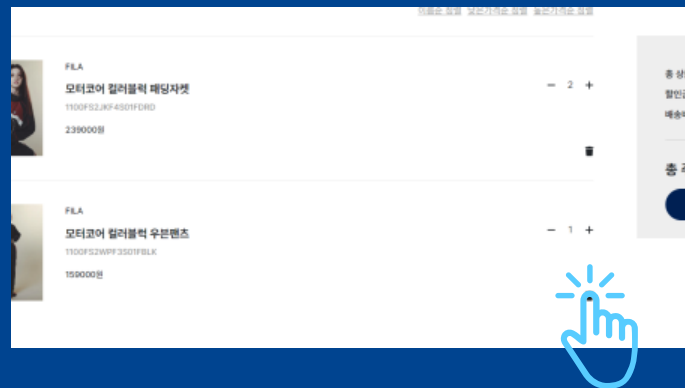
```
addItem(state, action){
  let num = state.findIndex(a => a.id == action.payload.id)
  let stateId = action.payload.id
  if(stateId === 0 || stateId === 2){
    if(action.payload.id === state[num].id){
      state[num].count++
    }
  }else{
    state.push(action.payload)
  }
},
```

React

```
<button onClick={() => {
  dispatch(addItem({
    id: moterCoreData[id].id, name: moterCoreData[id].name, price: moterCoreData[id].price
    , code: moterCoreData[id].code, count: 1, imgUrl: moterCoreData[id].imgUrl
  })))
  navigate('/cart')
}}>카트담기</button>
```

Delete item

휴지통 아이콘을 누르면 삭제 확인 창이 뜨고, 상품이 삭제됩니다.



Redux

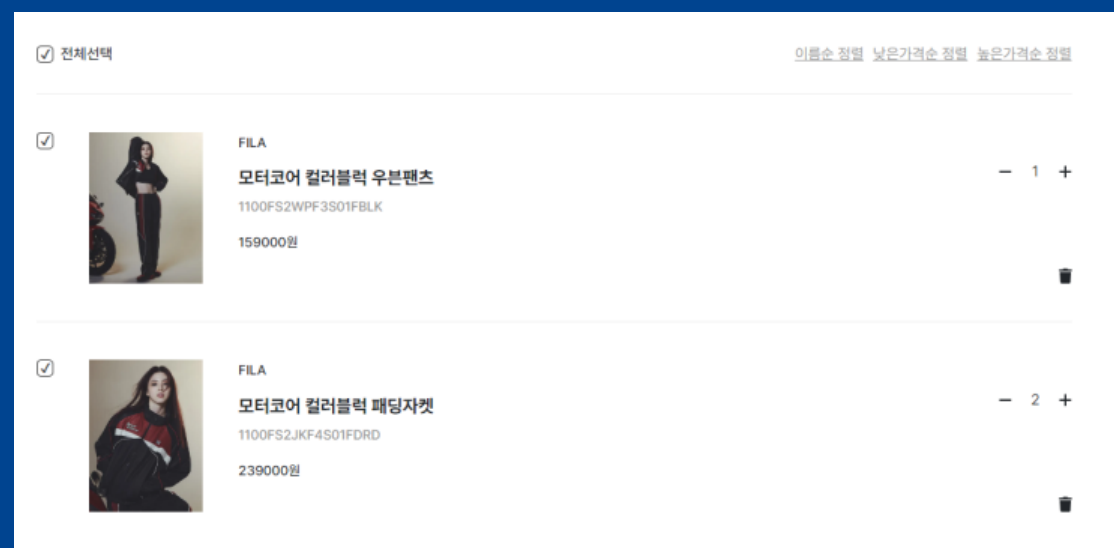
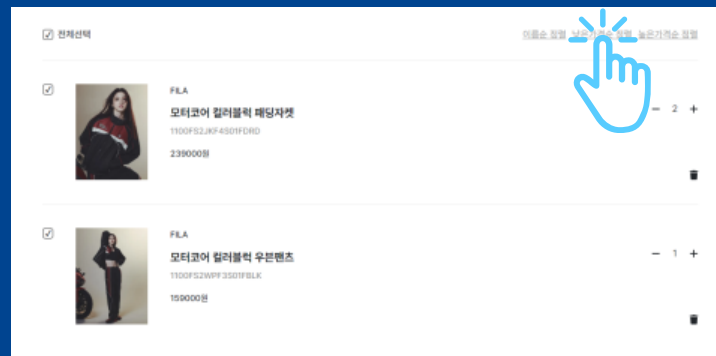
```
deleteItem(state, action){
  let num = state.findIndex(a => a.id == action.payload)
  alert(` ${state[num].name} ` 상품을 삭제하시겠습니까?`)
  state.splice(num,1)
},
```

React

```
<FontAwesomeIcon icon={faTrash} className='faTrash' style={{cursor:'pointer'}}
  |   onClick={()=>{dispatch(deleteItem(v.id))}}
  />
```

Sort item

상품 정렬 버튼을 누르면 각 항목에 맞게 정렬됩니다.



Redux

```
sortName(state){
  state.sort((a,b)=>a.name > b.name ? 1 : -1)
},

sortPrice_row(state){
  state.sort((a,b)=>a.price > b.price ? 1 : -1)
},

sortPrice_high(state){
  state.sort((a,b)=>a.price < b.price ? 1 : -1)
}
```

React

```
<div>
  <span onClick={()=>{dispatch(sortName())}}>이름순 정렬</span>
  <span onClick={()=>{dispatch(sortPrice_row())}}>낮은가격순 정렬</span>
  <span onClick={()=>{dispatch(sortPrice_high())}}>높은가격순 정렬</span>
</div>
```

평가 및 소감

처음 리액트 프로젝트를 시작했을 때는 처음 사용해보는 것들이 많아서 적응하기까지 시간이 오래 걸렸지만 완성되어 갈수록 익숙해지면서 또 하나의 기술을 배울 수 있어서 뿌듯했습니다.

새로운 기술들을 배우면서 전보다 더 많은 기능들을 사용할 수 있어서 좋았고, 앞으로 더 멋진 페이지를 만들기 위해서는 공부를 열심히 해야겠다고 느꼈습니다.

