

SPAO

Project 5 (기간 4/10 ~ 4/22)

SPAO

목차

- 1 주제 선정, 페르소나, 컬러 시스템
- 2 전체 UX, UI 이미지
- 3 코드 설명
- 4 느낀점

SPAO

1-1 주제 선정

스파오는 현재 많은 고객들에게 인기를 얻고 있다.

그 이유는,

1. 빠르게 변하는 의류 시장의 트렌드 흡수.

2. 데이터를 활용한 디테일을 통해 다양한 고객의 니즈를 충족.

이러한 그들의 경영 방식이 잘 드러난 것 같은, 깔끔하면서도
직관적인 정보를 제공하는 스파오 사이트를 첫 리액트
프로젝트로 제작해 보고 싶어서 주제로 선정하게 되었다.



1-2 페르소나



이름	성지수
나이	20세
관심사	패션 및 스타일링, 학교 생활 및 동아리 활동
특징	<p>대학생으로 학업에 바쁘지만 패션에도 관심이 많은 젊은이.</p> <p>캐주얼한 스타일을 선호하며, 트렌디하고 젊은 브랜드에 관심이 많음.</p>
요구사항	<p>스파오 웹사이트에서는 편안하고 스타일리시한 옷을 찾고,</p> <p>가격이 합리적이며 다양한 디자인의 제품을 쉽게 찾을 수 있는 것이 중요함.</p>

1-3 페르소나



이름	이준호
나이	30대 초반
관심사	패션, 트렌드, 온라인 쇼핑
특징	<p>회사원으로서 바쁜 일상을 보내고 있음, 패션에 대한 높은 관심을 가지고 있음. 멋진 스타일을 추구하며, 자주 온라인에서 쇼핑을 함. 특히, 편안하면서도 세련된 스타일을 선호함.</p>
요구사항	<p>스파오 웹사이트에서는 편안하면서도 스타일리시한 옷을 찾고, 다양한 옵션을 빠르게 비교하여 쇼핑할 수 있는 것이 중요함. 또한, 편리한 결제 및 배송 서비스를 기대 중임.</p>

1-4 Color System

A solid red square color swatch.

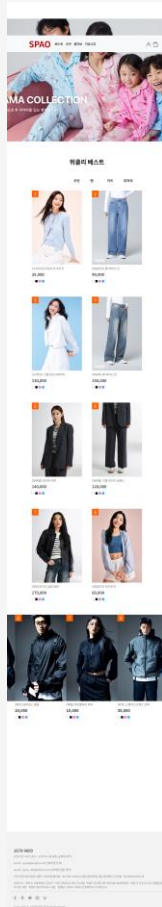
#bd0000

A white square color swatch with a thin black border.

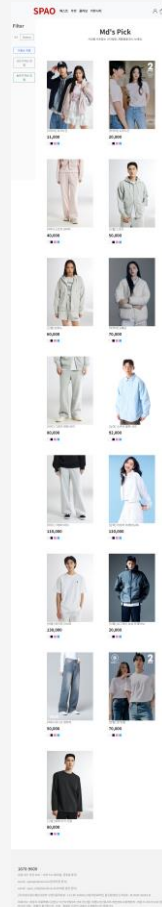
#ffff

2-1 전체 UX, UI 이미지

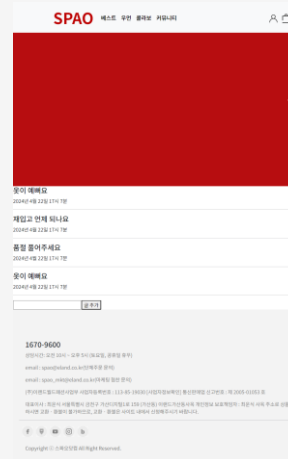
1 Home



2 Best Woman



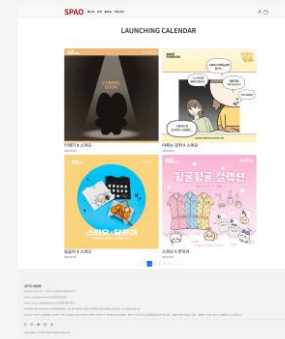
3 Community



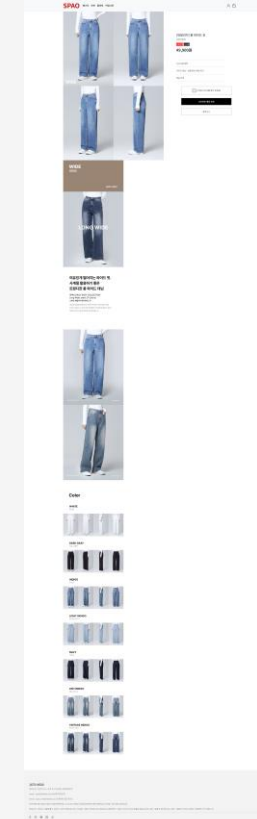
4 Login



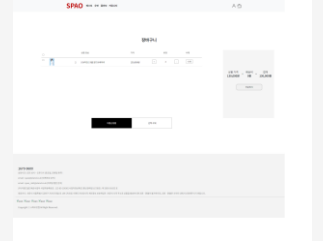
5 Colabo



6 Detail

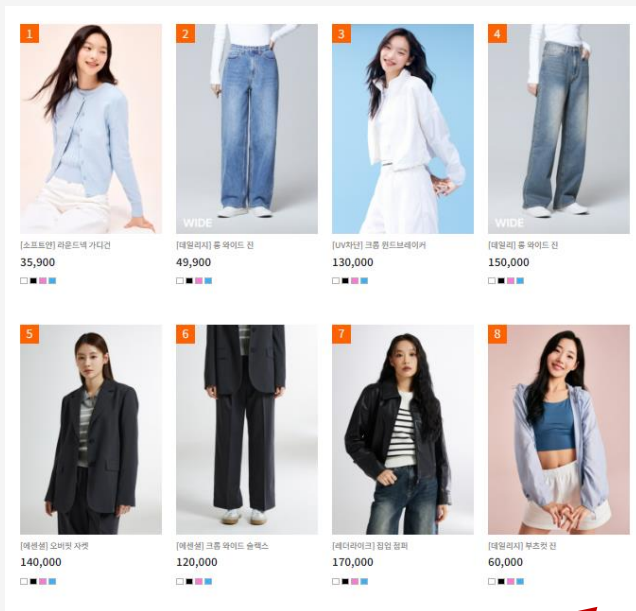


7 Cart



3-1 Home 코드 설명

Home tab



```
const Home = (props) => {  
  <h1>위클리 베스트</h1>  
  <Nav variant="pills" defaultActiveKey="/home" className="weeklybest">  
    <Nav.Item>  
      <Nav.Link onClick={() => { setTab(0) }} eventKey="link-0">우먼</Nav.Link>  
    </Nav.Item>  
    <Nav.Item>  
      <Nav.Link onClick={() => { setTab(1) }} eventKey="link-1">맨</Nav.Link>  
    </Nav.Item>  
    <Nav.Item>  
      <Nav.Link onClick={() => { setTab(2) }} eventKey="link-2">  
        키즈  
      </Nav.Link>  
    </Nav.Item>  
    <Nav.Item>  
      <Nav.Link onClick={() => { setTab(3) }} eventKey="link-3">  
        팔자마  
      </Nav.Link>  
    </Nav.Item>  
  </Nav>  
}
```

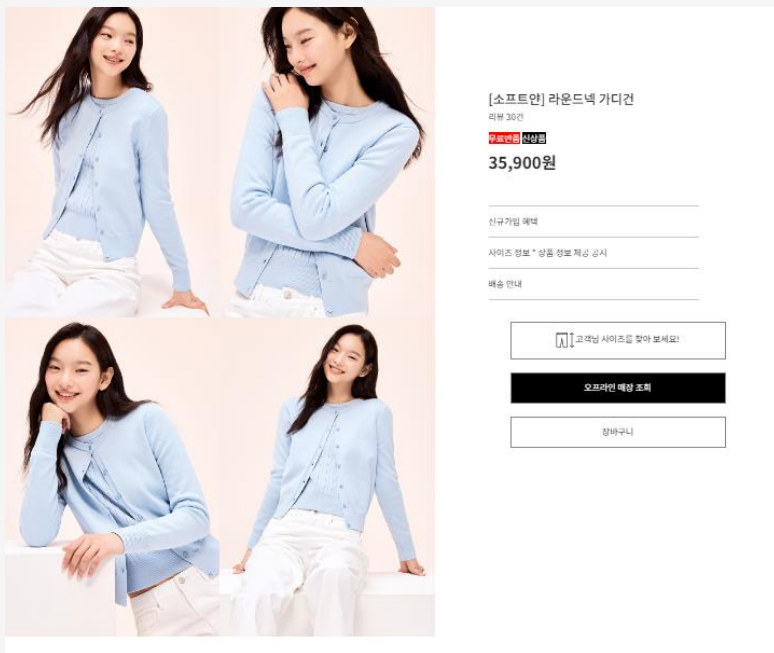
```
<HomeTabs tab = {tab} clothes = {clothes} no = {no}/>
```

```
const HomeTabs = (props) => {  
  let { tab, clothes, no } = props;  
  return (  
    <Container className="home_product_container">  
      <Row className="home_product">  
        {  
          clothes.slice(tab*8, (tab*8)+8).map((clo, i) =>  
            <Product clothes={clo} key={i} i={i} no={no} tab = {tab} />  
          )  
        }  
      </Row>  
    </Container>  
  )  
}
```

Clothes 데이터를
Slice로 나누고 map으로돌림

3-1 Home 코드 설명

Detail



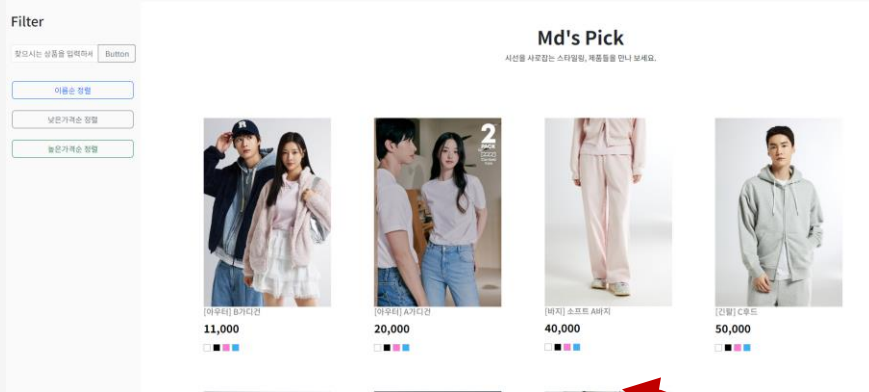
```
<>
<Nav.Link onClick = {(())=> navigate('/detail/' + no[i+(tab*8)])}>

  <Col md = {1} style={{ position: 'relative' }}>
    <span style={{ position: 'absolute', width: '40px', height: '40px', backgroundColor: '
    display: 'flex', justifyContent: 'center', alignItems: 'center' }}>{no[i+1]}</span>
    <img src = {imgUrl} width="80%" alt = "item" />
    <h4>{title}</h4>
    <p>{price.toLocaleString()}</p>
    <div className='color-squares'>
      <span></span>
      <span></span>
      <span></span>
      <span></span>
    </div>
  </Col>
</Nav.Link>
```

각각의 Product를 클릭하면, detail/no[i+(tab*8)] 페이지로 이동하고 tab 값이 변경될 때마다 새로운 데이터가 적용됨

3-2 Best, Woman 코드 설명

Best



```
<InputGroup className="mb-3">
  <Form.Control
    placeholder="찾으시는 상품을 입력하세요."
    aria-label="찾으시는 상품을 입력하세요."
    aria-describedby="basic-addon2"
    value={searchValue}
    onChange={handleSearchChange}
  />
  <Button variant="outline-secondary" id="button-addon2">
    Button
  </Button>
</InputGroup>
```

filterData를 통해
searchValue 값이 포함된
데이터를 따로 필터링 함.

```
<Button variant='outline-primary' onClick={() => {
  let copy = [...bestdata].sort((a, b) => (a.title > b.title) ? 1 : -1)

  setBestdata(copy);

  let no_copy = [];
  for (let i in copy) {
    no_copy.push(copy[i].id);
  }

  setNo(no_copy);
}}>이름순 정렬</Button>{' '}
```

```
<Button variant='outline-secondary' onClick={() => {
  let copy = [...bestdata].sort((a, b) => (a.price > b.price) ? 1 : -1)

  setBestdata(copy);

  let no_copy = [];
  for (let i in copy) {
    no_copy.push(copy[i].id);
  }

  setNo(no_copy);
}}>낮은가격순 정렬</Button>{' '}
```

```
<Button variant='outline-success'
  onClick={() => {
    let copy = [...bestdata].sort((a, b) => (a.price < b.price) ? 1 : -1)

    setBestdata(copy);

    let no_copy = [];
    for (let i in copy) {
      no_copy.push(copy[i].id);
    }

    setNo(no_copy);
  }}>높은가격순 정렬</Button>
```

이름 순 정렬 버튼

[...bestData] 타이틀 순으로
정렬.

setNo에 no_copy 배열을
붙여넣기.

낮은 가격순, 높은 가격순도
마찬가지.

3-3 Login 코드 설명

```
import './styles.scss',

const Login = () => {
  const [id, setId] = useState('');
  const [password, setPassword] = useState('');

  const handleIdChange = (e) => {
    setId(e.target.value);
  };

  const handlePasswordChange = (e) => {
    setPassword(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();

    chkID();
    chkPW();
  };

  const chkPW = () => {
    // 비밀번호 체크 로직
    var reg = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[!@#$%^&*~]).{8,}$/;
    var hangulcheck = /[ㄱ-ㅎ|ㅏ-ㅣ|가-힣]/;

    if (!reg.test(password)) {
      alert('비밀번호는 8자 이상이어야 하며, 숫자/대문자/소문자/특수문자를 모두 포함해야 합니다.');
```

→

지금 로그인하고 다양한 혜택을 만나 보세요

ID

We'll never share your email with anyone else.

Password

☐ Check me out



handleSubmit으로
Form 기본 동작을 막고,
chkID(), chkPW()를 실행시켜서 로그인
형식 검사함

3-4 Colabo 코드 설명

```
/* Pagination */
<Row>
  <Pagination className="pagination justify-content-center" size="md">
    <Pagination.First onClick={() => handlePaginationClick(1)} />
    <Pagination.Prev onClick={() => handlePaginationClick(currentPage - 1)} disabled={currentPage === 1} />
    {Array.from({ length: totalPages }, (_, i) => {
      <Pagination.Item key={i + 1} active={i + 1 === currentPage} onClick={() => handlePaginationClick(i + 1)}>
        {i + 1}
      </Pagination.Item>
    })}
    <Pagination.Next onClick={() => handlePaginationClick(currentPage + 1)} disabled={currentPage === totalPages} />
    <Pagination.Last onClick={() => handlePaginationClick(totalPages)} />
  </Pagination>
</Row>
</Col>
)
```



터레기 X 스파오

2023.05.06



마루는 강쥐 X 스파오

2023.04.25



담곰이 X 스파오

2023.04.12



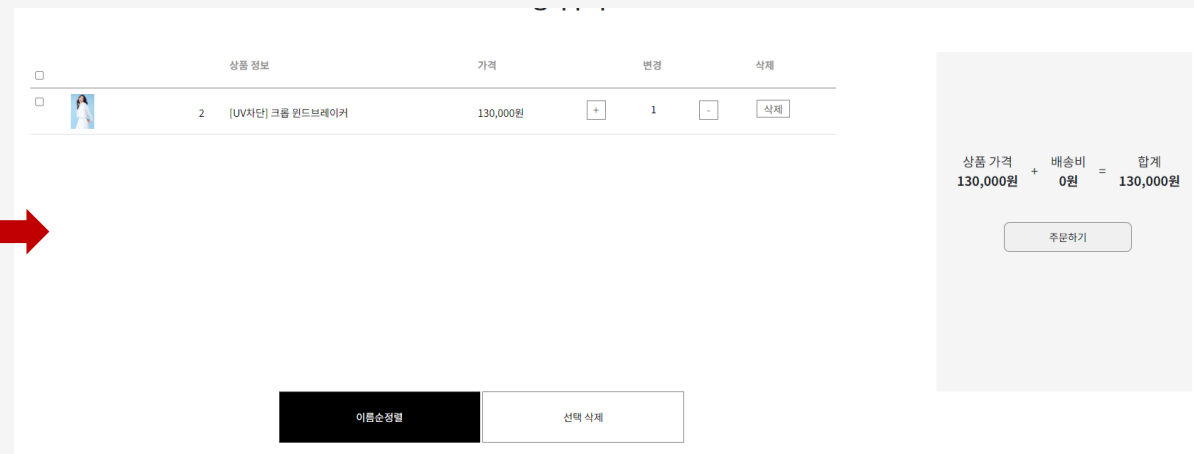
스파오 X 먼작귀

2023.03.31

<, > 버튼 누르면 한 칸씩 page 앞 뒤로 이동.
첫 페이지, 마지막 페이지일 때는 동작 막음.
<< 버튼 누르면 맨 앞 page로 이동
>> 버튼 누르면 맨 끝 page로 이동

3-5 Cart 코드 설명

```
addCount(state, action) {  
  const { payload: newItem } = action;  
  const itemIndex = state.findIndex(item => item.id === newItem.id);  
  
  if (itemIndex > -1) {  
    state[itemIndex].amount += 1;  
    alert('이미 상품이 담겨 있습니다. 수량이 합산됩니다.');
```



장바구니에 상품이 이미 담겨져 있는 경우,
해당 상품의 수량을 증가시키고 alert을 띄움

그렇지 않은 경우는 상품을 장바구니에 추가함.

4-1 느낀점

1. 리액트 컴포넌트 개념에 대해 헷갈린 상태였는데 직접 첫 리액트 프로젝트를 제작해 보면서 컴포넌트와 라우터를 어떻게 분리하는지 개념을 정리할 수 있어서 좋았다.
2. useState, useEffect 등... 각각 리액트의 Hook 함수에 대해 공부할 수 있어서 좋았고, 리액트를 사용하니 웹페이지를 훨씬 간편하고 쉽게 만들 수 있다는 생각이 들었다.
3. 자바스크립트 부분이 아직 부족하다는 것을 깨달았다. Best 페이지에서 가격 별로 상품을 분류하는 필터링을 구현하고 싶었는데 못 했다. 자바스크립트 개념을 더 공부하여 잘 활용할 수 있도록 노력해야겠다.

감사합니다

SPAO