

PROJECT 5

A close-up photograph of a woman's back and shoulder. She is spraying a clear liquid from a small, rectangular glass perfume bottle with a gold cap. A fine mist of the perfume is visible as it sprays out. The woman has long, dark hair tied back in a ponytail.

# JOVY

YU HYE LIM

THE EMBASSY OF  
RARE PERFUMES

# CONTENTS



기획

주제선정

페르소나

일정표

화면설계

와이어프레임

프로토타입

코드구현

Main Page

Sub Page

평가/소감

# *Brain Storming*





# PERSONA

## *LIFE STYLE*

청순한 이미지로 사랑 받고 있는 미혼의 여배우.

밖에 돌아다니면 사람들이 알아보는 게 힘들어, 어쩌다 보니 일 할 때 외에는 대부분 집에서 시간을 보낸다. 꾸미는 게 직업인 만큼 치장하는 것에 관심이 많고, 향수도 예외는 아니다.

보통은 백화점에 가서 향을 맡아보고 구입하지만, 이번만큼은 시간에 구애 받지 않고 구매할 수 있는 인터넷으로 향수를 구매해보고자 한다.

## *GOAL*

- 향을 직접 맡아보지 못하므로 향에 대한 구체적인 설명과 함께 향이 자세하게 묘사되어 있으면 좋겠다.
- 원하는 향을 먼저 고르면 그것과 비슷한 향을 추천해주었으면 좋겠다.
- 너무 유명한 브랜드보다는 많이 알려지지 않은 브랜드의 향수가 다양하게 있으면 좋겠다.

이름 : 유아진

나이 : 31세

직업 : 영화 배우

사는 곳 : 서울시

성격 : 차분하고 조용한 성격



이름 : 박진희

나이 : 22세

직업 : 국제학부 대학생

사는 곳 : 서울시

성격 : 발랄하고 귀여운 성격

# PERSONA

## LIFE STYLE

발랄하고 애교 많은 성격의 소유자로, 같은 과 선배와 연애를 시작한 지 1년이 다 되어간다. 곧 있을 남자친구와의 1주년을 기념하기 위해, 선물을 고민하다가 향수를 선물하기로 했다. 그래서 잘 알려지지 않았지만 아는 사람들만 안다는 JOVOY에서 향수를 구입하기로 한다. 남자 향수는 처음 구매하는 것이라 남자 친구가 과연 선물을 좋아할지 걱정이 많아, 남자들에게 좋은 평을 듣는 향수를 고르고자 한다.

## GOAL

- 성별에 따른 리뷰가 나뉘어 있으면 좋겠다.
- 남자들에게 인기가 많은 향수인지 알기 위해, 구매자의 연령 별, 성 별 구매 통계를 볼 수 있으면 좋겠다.
- 가격이 너무 비싸지 않은 향수를 우선적으로 보고 싶다.



# IMELINE

**3.29**

주제 선정 및 기획

**4.1 - 4.22**

코드 구현

**4.23**

마무리

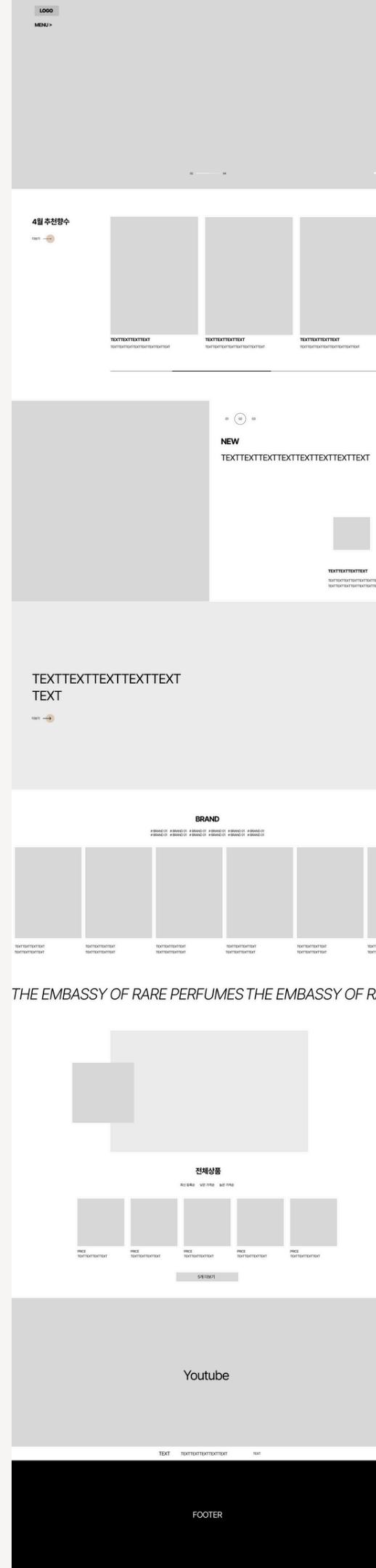


# WIRE FRAME



Cart page ▲

Main page ▲



Youtube

FOOTER

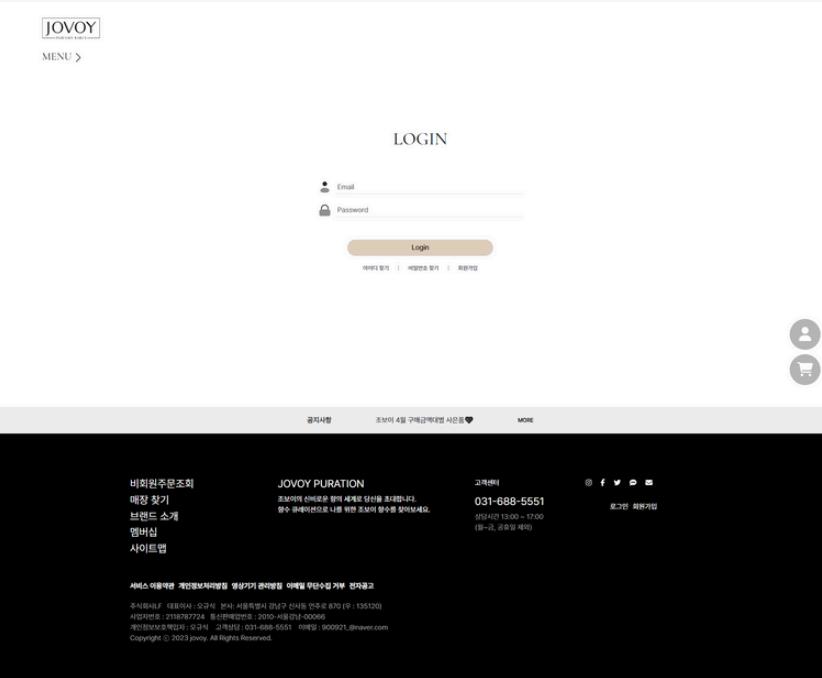
Detail page ▼



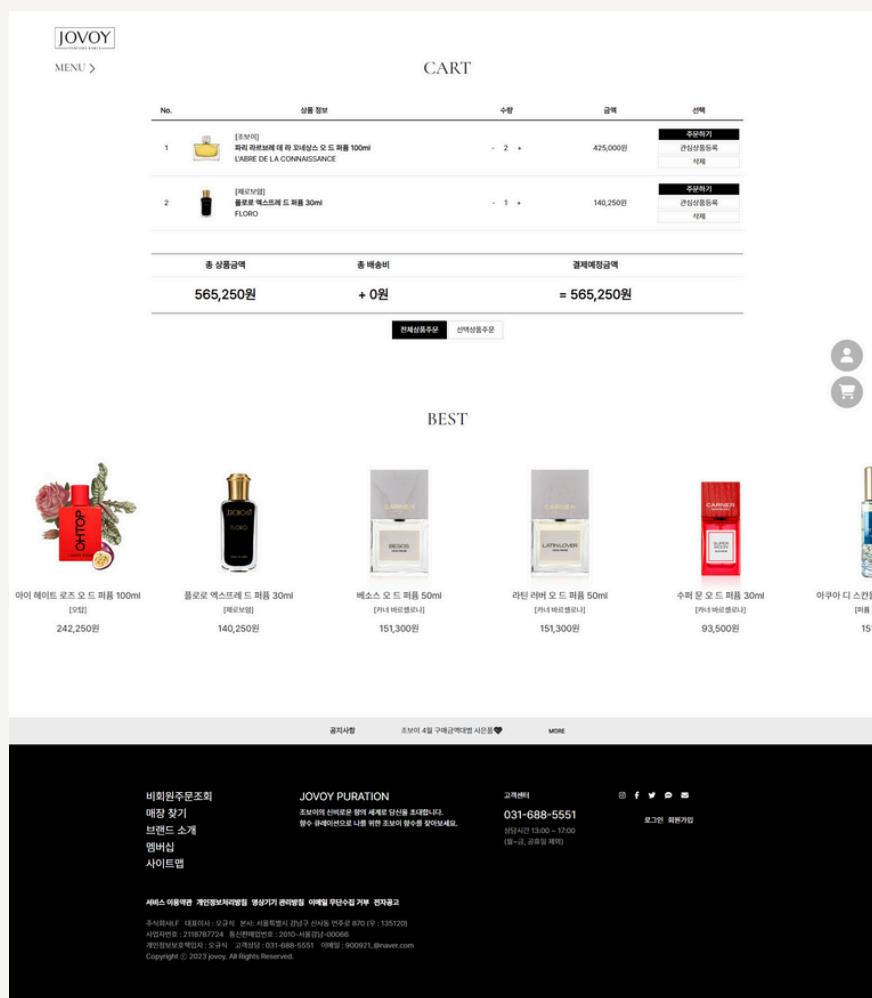
FOOTER

# PROTO TYPE

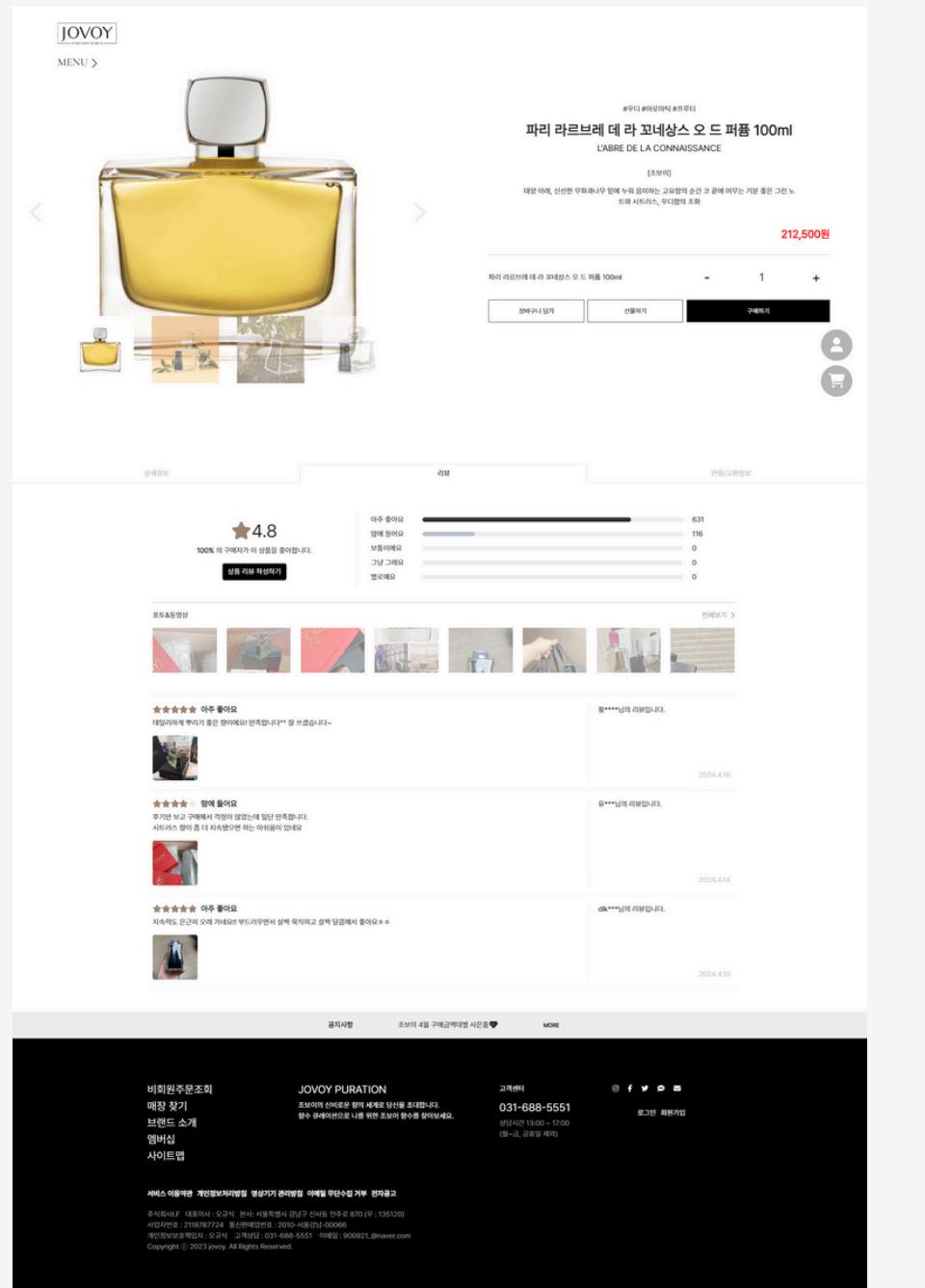
# ▼ Login page



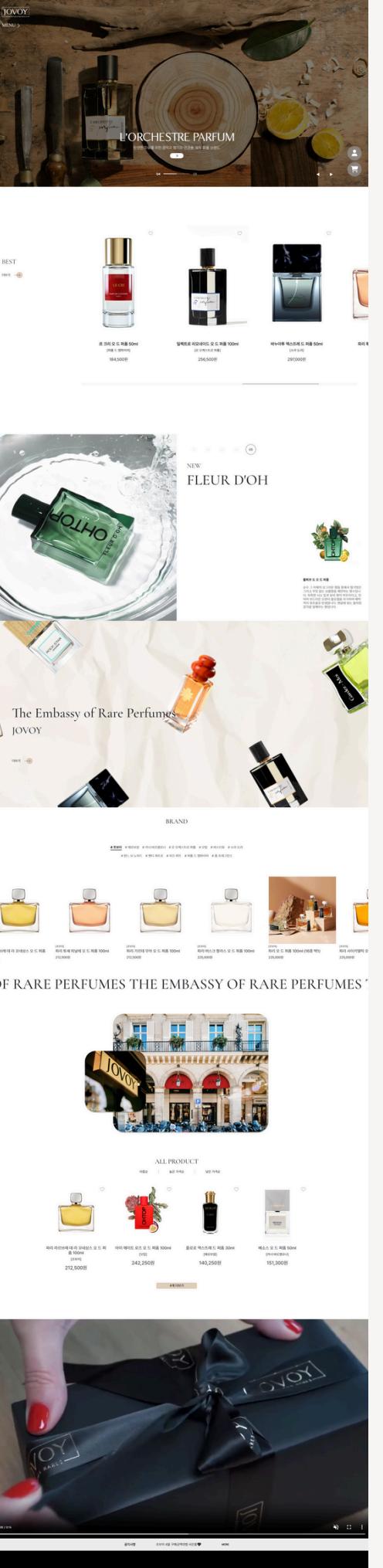
## ▲ Cart page



# ▲ Detail page



200

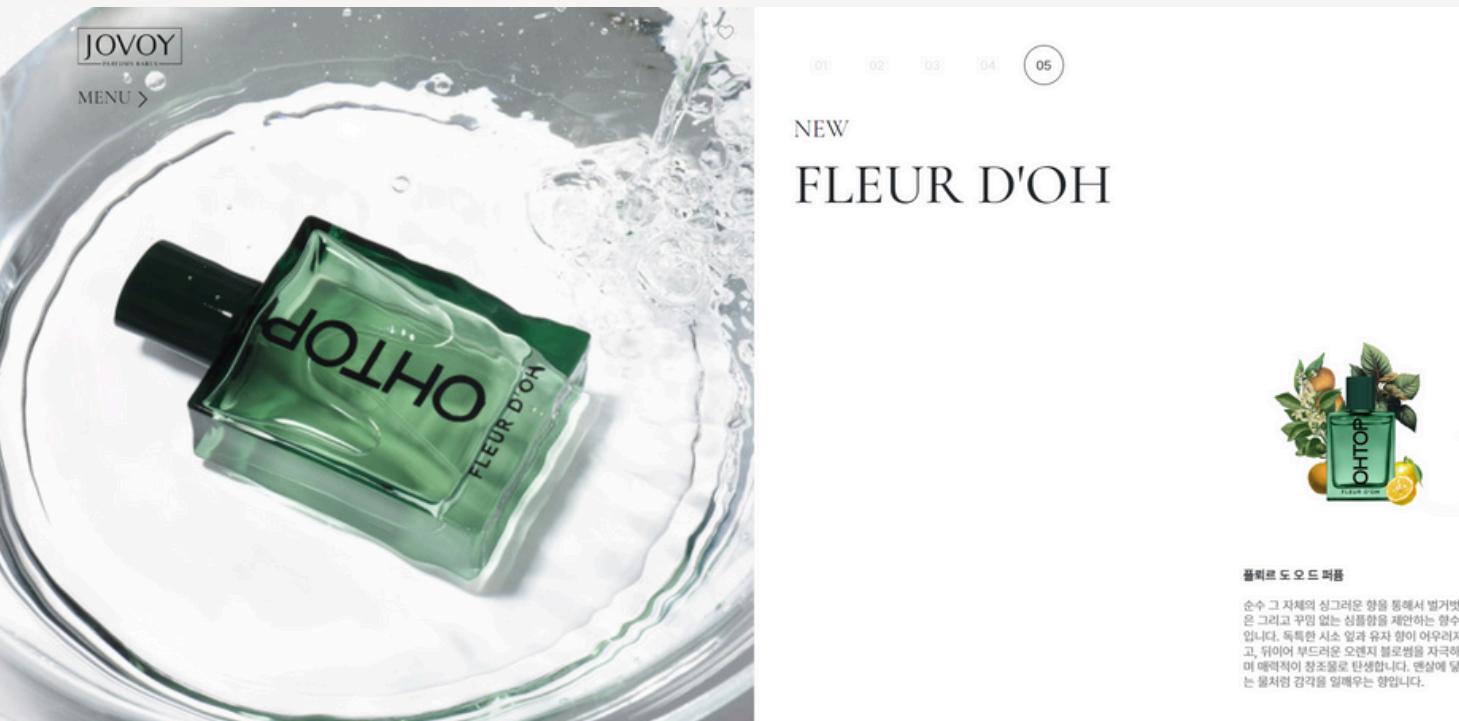


# CODE

# MAIN PAGE

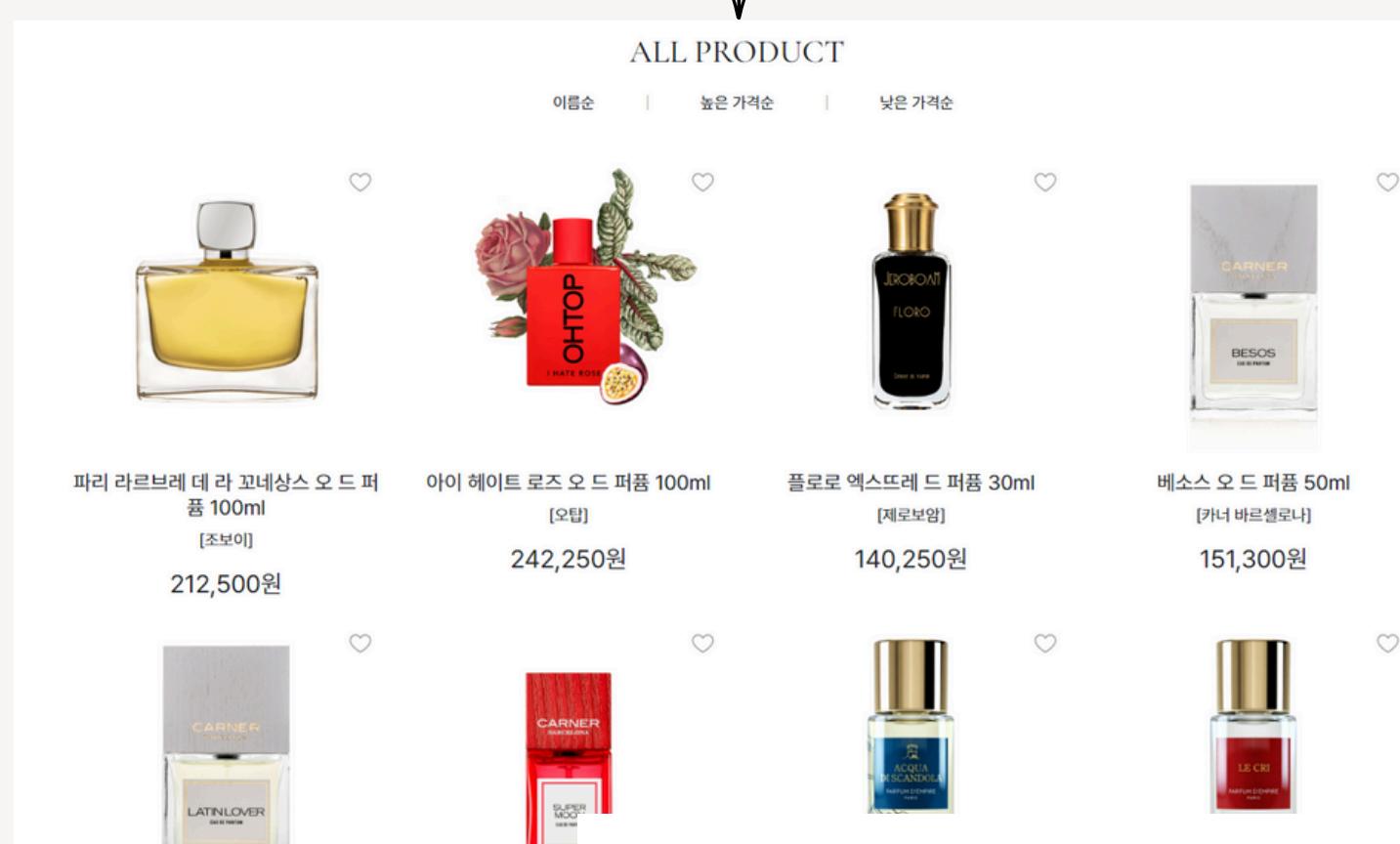
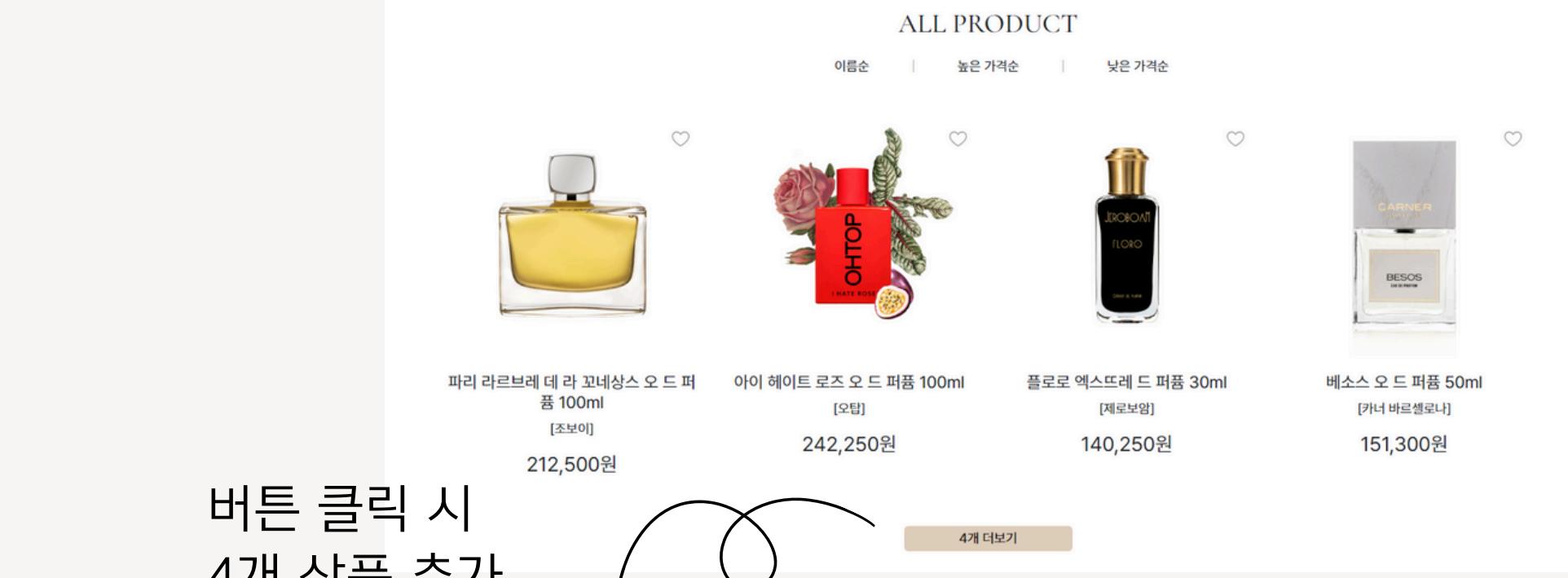
Swiper Slider를 이용하여 슬라이드를 구현했습니다.  
Controller를 설정하여 2개의 슬라이드를 연동하였고,  
슬라이드 이동 시 다른 슬라이드도 함께 움직이도록 구현하였습니다.

app.js에서 props를 받아와서 데이터를 연결하고  
map()을 이용하여 반복되는 코드를 줄였습니다.



```
export default (props) => {
  // let [news] = useState(data2);
  let { news, no, i } = props;
  const [firstSwiper, setFirstSwiper] = useState(null)
  const [secondSwiper, setSecondSwiper] = useState(null)
  var bullet = ['01', '02', '03', '04', '05']
  let navigate = useNavigate();

  return (
    <div className="newWrap">
      <Swiper className="main_slider bigImg"
        modules={[Pagination, A11y, Controller, Autoplay]}
        autoplay={{ delay: 3000 }}
        loop={true}
        slidesPerView={1}
        onSwiper={setFirstSwiper}
        controller={{ control: secondSwiper }}>
        {
          news.map((perfume, i) =>
            <SwiperSlide key={i}>
              <img src={process.env.PUBLIC_URL + perfume imgUrlB} onClick={() => navigate('/detail/new/' + no[i])} alt="product" />
              <Wish />
            </SwiperSlide>
          )
        }
      </Swiper>
      <div className="boxR">
        <p className='page'></p>
        <h2>NEW</h2>
        <Swiper className='sub_slider'
          modules={[Pagination, A11y, Controller]}
          slidesPerView={1}
          loop={true}
          onSwiper={setSecondSwiper}
          controller={{ control: firstSwiper }}
          pagination={{
            clickable: true,
            el: '.page',
            clickable: 'true',
            renderBullet: function (index, className) {
              return <div class=' + className + '><span>' + (bullet[index]) + '</span></div>';
            }
          }}>
          {
            news.map((perfume, i) =>
              <SwiperSlide onClick={() => navigate('/detail/new/' + no[i])} key={i}>
                <h1>{perfume.titleE}</h1>
                <div className="smallbox">
                  <img src={process.env.PUBLIC_URL + perfume imgUrl} alt="imgS" />
                  <p>{perfume.title}</p>
                  <h6>{perfume.content}</h6>
                </div>
              </SwiperSlide>
            )
          }
        </Swiper>
      </div>
    </div>
  );
};
```



json파일 만들어서 git서버에 올린 후, axios 라이브러리 사용하여 서버에서 데이터 요청했습니다. 버튼을 클릭하면 4개의 상품이 추가 되도록 하였습니다. 또한 정렬 버튼을 클릭하면 상품 배열을 sort하여 이름 순, 가격 순으로 정렬할 수 있도록 하였습니다.

```

const AllProduct = (props) => {
  let navigate = useNavigate();
  const original = dataAll1;
  let { perfumes, setPerfumes, no, count, setCount } = props;

  return (
    <div className='AllWrap'>
      <h2>ALL PRODUCT</h2>

      <nav>
        <ul>
          <li onClick={() => {
            let copy = [...perfumes].sort((a, b) => (a.title > b.title) ? 1 : -1)
            setPerfumes(copy);
          }}>이름순</li>
          <li onClick={() => {
            let copy = [...perfumes].sort((a, b) => (a.price < b.price) ? 1 : -1)
            setPerfumes(copy);
          }}>낮은 가격순</li>
          <li onClick={() => {
            let copy = [...perfumes].sort((a, b) => (a.price > b.price) ? 1 : -1)
            setPerfumes(copy);
          }}>높은 가격순</li>
        </ul>
      </nav>

      <Container>
        <Row>
          {perfumes.map((perfume, i) =>
            <Col lg={3} md={4} sm={6} key={i} className='product' onClick={() => navigate('/detail/all/' + no[i])}>
              <div>
                <img src={process.env.PUBLIC_URL + perfume imgUrl} alt="" />
                <h5>{perfume.title}</h5>
                <p>{perfume.brand}</p>
                <h4>{perfume.price.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",") + "원"}</h4>
              </div>
              <Wish />
            </Col>
          )}
        </Row>
      </Container>

      <button onClick={() => {
        if (count === 1) {
          axios.get("https://hxelim.github.io/react_data/jovoy_all2.json").then(result) => {
            let copy = [...perfumes, ...result.data]
            setPerfumes(copy)
            setCount(2)
          }
        } else if (count === 2) {
          axios.get("https://hxelim.github.io/react_data/jovoy_all3.json").then(result) => {
            let copy = [...perfumes, ...result.data]
            setPerfumes(copy)
            setCount(3)
          }
        } else if (count === 3) {
          axios.get("https://hxelim.github.io/react_data/jovoy_all4.json").then(result) => {
            let copy = [...perfumes, ...result.data]
          }
        }
      }}>4개 더보기</button>
    </div>
  )
}

```

# CODE

# CART PAGE

No.	상품 정보	수량	금액	선택
1	[조보이] 파리 라르브레 데 라 고페상스 오 드 퍼퓸 100ml L'ABRE DE LA CONNAISSANCE	- 2 +	425,000원	<button>주문하기</button> <button>관심상품등록</button> <button>삭제</button>
2	[제로보인] 플로로 엑스프레 드 퍼퓸 30ml FLORO	- 1 +	140,250원	<button>주문하기</button> <button>관심상품등록</button> <button>삭제</button>
3	[오답] 아이 헤이트 로즈 오 드 퍼퓸 100ml I HATE ROSE	- 1 +	242,250원	<button>주문하기</button> <button>관심상품등록</button> <button>삭제</button>
총 상품금액		총 배송비	결제예정금액	
807,500원		+ 0원	= 807,500원	
		<button>전체상품주문</button>	<button>선택상품주문</button>	

Redux를 이용하여 Cart 페이지를 구현하였습니다. Detail 페이지에서 상품을 선택 후 구매 버튼을 누르면 Cart 페이지에 상품이 추가되고, 총 상품 금액이 계산될 수 있도록 구현하였습니다.

만일 Cart 페이지에 이미 있는 상품을 추가할 경우 수량만 추가되게 하여 목록에 중복이 생기지 않도록 하였습니다. 삭제 버튼을 누르면 목록에서 삭제되도록 구현하였습니다.

```

const Detail = (props) => {
  let { perfumes } = props;
  let { id } = useParams();
  let [ orderCount, setOrderCount ] = useState(1)
  let dispatch = useDispatch()
  const navigate = useNavigate()

  const moveCart = () => {
    dispatch(addItem({
      id: perfumes[id].id,
      imgUrl: perfumes[id].imgUrl,
      brand: perfumes[id].brand,
      id: perfumes[id].id,
      item: perfumes[id].title,
      titleE: perfumes[id].titleE,
      amount: orderCount,
      price: perfumes[id].price,
    }))
    navigate('/cart')
  }

  return (
    <div>
      <div className="detail">
        <Row>
          <Col sm={12} md={6} className='imgBox'>
            <DetailSlider perfumes={perfumes} id={id} />
            {/* <img src={perfumes[id].imgUrl} alt="item" /> */}
          </Col>
          <Col sm={12} md={6} className='explainBox'>
            <div className="explain">
              <p>{perfumes[id].scent}</p>
              <h2 className="title">{perfumes[id].title}</h2>
              <h5>{perfumes[id].titleE}</h5><br />
              <p>{perfumes[id].brand}</p>
              <p className="content">{perfumes[id].content}</p>
              <h4 className="price">{perfumes[id].price.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",") + "원"}</h4>
            </div>
            <hr width='100%' style={{ margin: 0 }} />
            <div className="count">
              <span>{perfumes[id].title}</span>
              <div className="countbox">
                <button className='minus' onClick={()=>{
                  setOrderCount(orderCount-1)
                  if (orderCount<=1){
                    alert('최소 주문 수량은 1개 입니다.')
                    setOrderCount(1)
                  }
                }}>-</button>
                <div id="result">{orderCount}</div>
                <button className='plus' onClick={()=>{
                  setOrderCount(orderCount+1)
                }}>+</button>
              </div>
            </div>
            <div className="buy">
              <button onClick={moveCart}>장바구니 담기</button>
              <button>선택하기</button>
              <button onClick={moveCart}>구매하기</button>
            </div>
          </Col>
        </Row>
      </div>
    </div>
  )
}

```

# CODE

# LOGIN PAGE

The screenshot shows a login form with the following elements:

- Email input field with placeholder "Email" and a user icon.
- Password input field with placeholder "Password" and a lock icon.
- A large orange "Login" button.
- Links at the bottom: "아이디 찾기" (Find ID), "비밀번호 찾기" (Find Password), and "회원가입" (Sign Up).

정규표현식을 이용하여  
로그인 페이지를 구현하였습니다.

```
const Login = () => [
  const id = useRef()
  const password = useRef();

  function pass() {
    const idIv = id.current.value;
    const passIv = password.current.value;
    const emailRegEx = /^(?:[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,6})$/;
    const reg = /^(?=.*[a-zA-Z])(?=.*[a-zA-Z])(?=.*[0-9])(?=.*[$%^&*]).{8,}$/;

    if (idIv === '') {
      alert('이메일 주소를 입력해주세요.');
      id.current.focus();
    } else {
      if (!emailRegEx.test(idIv)) {
        alert('이메일을 다시 입력해 주세요');
        id.current.focus();
        return false;
      } else {
        if (false === reg.test(passIv)) {
          alert('비밀번호는 8자 이상이어야 하며, 숫자/대문자/소문자/특수문자를 모두 포함해야 합니다.');
          password.current.focus();
          return false;
        } else if (passIv.search(/\s/) !== -1) {
          alert('비밀번호는 공백없이 입력해주세요.');
          password.current.focus();
          return false;
        } else {
          alert('환영합니다!');
        }
      }
    }
  }

  return (
    <div className='loginWrap'>
      <Container>
        <h1>LOGIN</h1>
        <div className='form'>
          <div className='email'>
            <svg xmlns='http://www.w3.org/2000/svg' width='1em' height='1em' viewBox='0 0 24 24'><circle cx='12' cy='6' r='4' fill='currentColor' /><path fill='currentColor' d='M20 17.5c0 2.485 0 4.5-8 4.5s-8-2.015-8-4.5S7.582 13 12 13s8 2.015 8 4.5' opacity='0.5' /></svg>
            <input type='text' name='mail' placeholder=' Email' ref={id} />
          </div>
          <div className='pw'>
            <svg xmlns='http://www.w3.org/2000/svg' width='1em' height='1em' viewBox='0 0 24 24'><path fill='currentColor' d='M2 16c0-2.828 0-4.243 0-5.121 0-5.121 0-5.121 0-5.121C3.757 10 5.172 10 8 10h8c2.828 0 4.243 0 5.121 0 5.121 0 5.121 0 5.121 0 5.121C22 11.757 22 13.172 22 16c0 2.828 0 4.243-0.5 5.121-0.5 5.121-0.5 5.121-0.5 5.121C20.243 2 18.828 2 16 22H8c-2.828 0-4.243 0-5.121-0.5 5.121-0.5 5.121-0.5 5.121-0.5 5.121C18.228 22 16 22 16 22' opacity='0.5' /><path fill='currentColor' d='M6.75 8a5.25 5.25 0 0 1 10.5 0v2.004c.567.005 1.064.018 1.5.05V8a6.75 6.75 0 0 0-13.5 0v2.055a23.57 23.57 0 0 1 1.5-.051z' /></svg>
            <input type='text' name='pw' placeholder=' Password' ref={password} />
          </div>
          <button onClick={pass}>Login</button>
          <ul>
            <li>아이디 찾기</li>
            <li>비밀번호 찾기</li>
            <li>회원가입</li>
          </ul>
        </div>
      </Container>
    </div>
  )
}
```



# REVIEW

React를 사용하여 처음 프로젝트를 시작할 때에는 너무 막막했습니다. 코드가 익숙하지 않고 구조가 머리 속에 그려지지 않아 어디부터 시작해야 할지 감이 잘 잡히지 않았습니다. 그래서 하나의 데이터 파일을 내려받아 작성해도 될 코드를 구역마다 데이터 파일을 나누어 만듦으로서 코드가 더 복잡해지고, 변경 사항이 있을 때마다 각각의 데이터 파일을 수정해야 하는 등의 시행착오를 겪기도 했습니다.

아직 React를 익숙하게 사용하지는 못하지만, 이번 프로젝트를 통해 다양하게 사용해보며 하나하나 원하는 기능들을 구현해가니 React 라이브러리의 편리함을 느낄 수 있었습니다. 앞으로 있을 React 프로젝트에서는 이번의 경험을 토대로 React의 장점을 최대한 활용하여 간단하고 알기 쉬운 코드를 작성할 수 있도록 노력할 것입니다.

# THANK YOU

