
PORTFOLIO

#React - Apple Store

By_ Kim Ga Ram

CONTNENS

1. 주제선정

- 주제선정

2. 기획

- 페르소나
- 와이어 프레임
- STYLE SYSTEM
- 프로토타입

3. 구현

- 코드구현

4. 마무리

- 평가 및 후기

06

TIMELINE

2025

S U N	M O N	T U E	W E D	T H U	F R I	S A T
1	주제선정 및 Main 디자인	2	3	4	5	6
7	Main 코드 구현					
8	9	10	11	12	13	14
		sub 코드 구현				
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

1. 주제선정

Apple Store

익숙하고 직관적인 Apple Store의 재해석

기존 애플 스토어 웹사이트는 방대한 정보와 복잡한 인터페이스로 인해,
사용자가 원하는 정보를 빠르게 찾기 어렵고 조작 과정에서 혼란을 느낄 수 있다고 생각했습니다.
특히 처음 방문하는 사용자에게는 제품 탐색이나 구매 절차가 직관적이지 않아 접근성이 떨어진다고 판단했습니다.

이에 따라 사용자의 실제 사용 흐름에 맞춰,
더 간결하고 친숙한 UI/UX 구조로 애플 스토어를 재구성하고자 기획하게 되었습니다.

핵심 제품 정보에 집중하면서도, 불필요한 요소를 줄이고,
누구나 쉽게 조작할 수 있는 환경을 통해 사용성을 높이는 데 중점을 두었습니다.



2. 기획 - 페르소나



이름: 이준혁

나이: 38세

직업: IT 기업 기획팀 과장

거주지: 경기도 성남

디지털 숙련도: 보통 이상

- 기기 사용: MacBook Pro, iPhone 15 Pro, Apple Watch
- 구매 성향: 빠르고 정확하게 핵심 정보만 확인 후 결제
- 목표: 신제품 정보를 빠르게 비교하고, 불필요한 단계 없이 제품 구매

Pain Points

- 페이지는 이쁘나,
정보를 한눈에 보기 어려움
- 제품 탐색하기 어려움
- 구매 절차가 직관적이지 않음

User Needs

- 간단하고 빠른 제품 비교
- 불필요한 홍보 콘텐츠 최소화
- 명확한 제품 스펙 및 가격 확인
- 직관적인 구매 흐름

2. 기획 - 페르소나



이름: 김서현

나이: 28세

직업: 콘텐츠크리에이터

거주지: 서울 마포구

디지털 숙련도: 높음 (디자인/편집 툴 활용)

- 기기 사용: iPad Pro, MacBook Air, AirPods Pro
- 구매 성향: 제품 기능과 디자인을 비교한 후, 리뷰와 정보 확인에 시간 많이 씀
- 목표: 자신의 콘텐츠 제작에 맞는 최적의 기기를 효율적으로 찾는 것

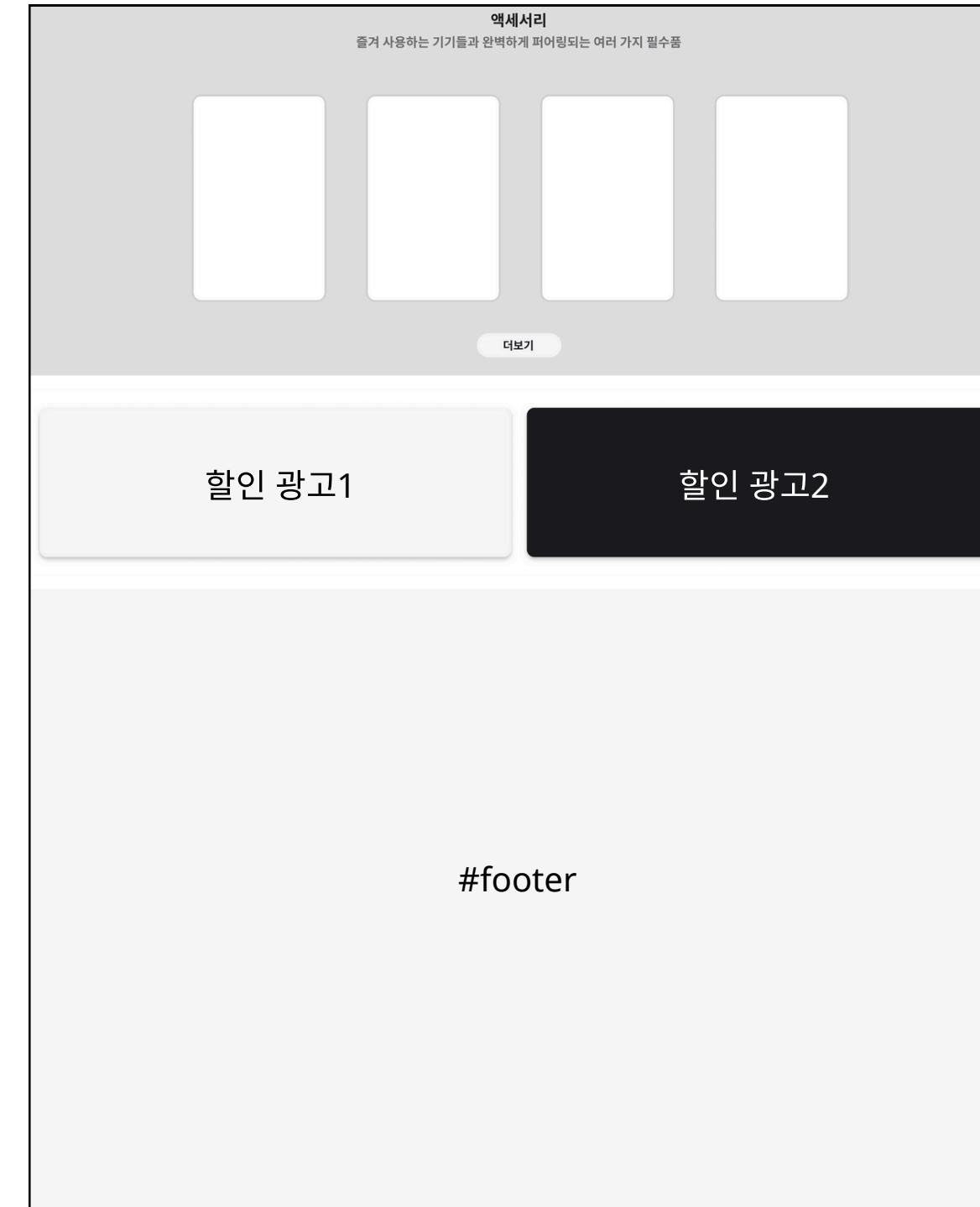
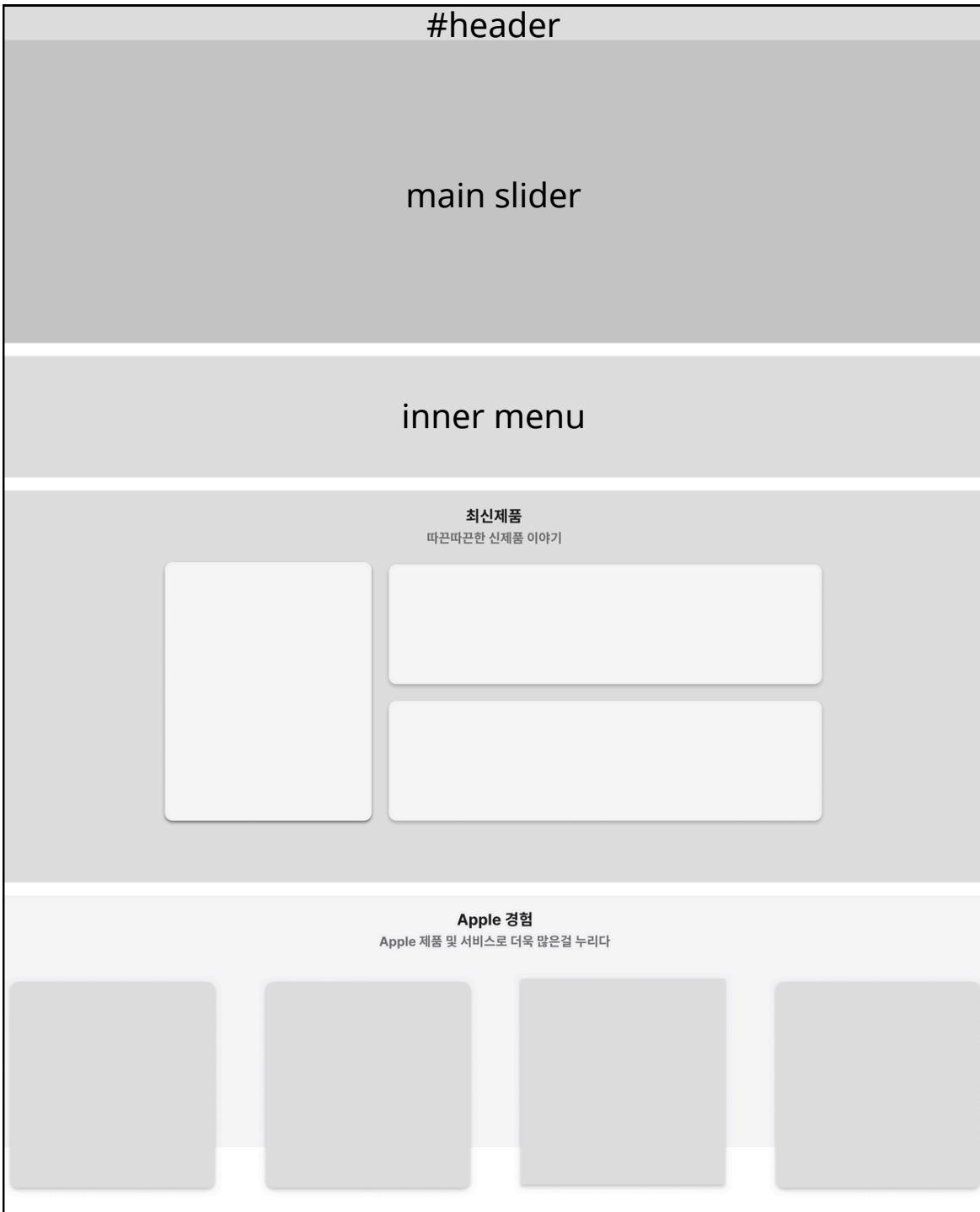
Pain Points

- 홈페이지는 예쁘지만, 글로벌 메뉴 구성으로 원하는 정보를 한번에 찾기 어렵다.
- 추천 제품이 광고 위주로 구성되어 신뢰도가 낮다

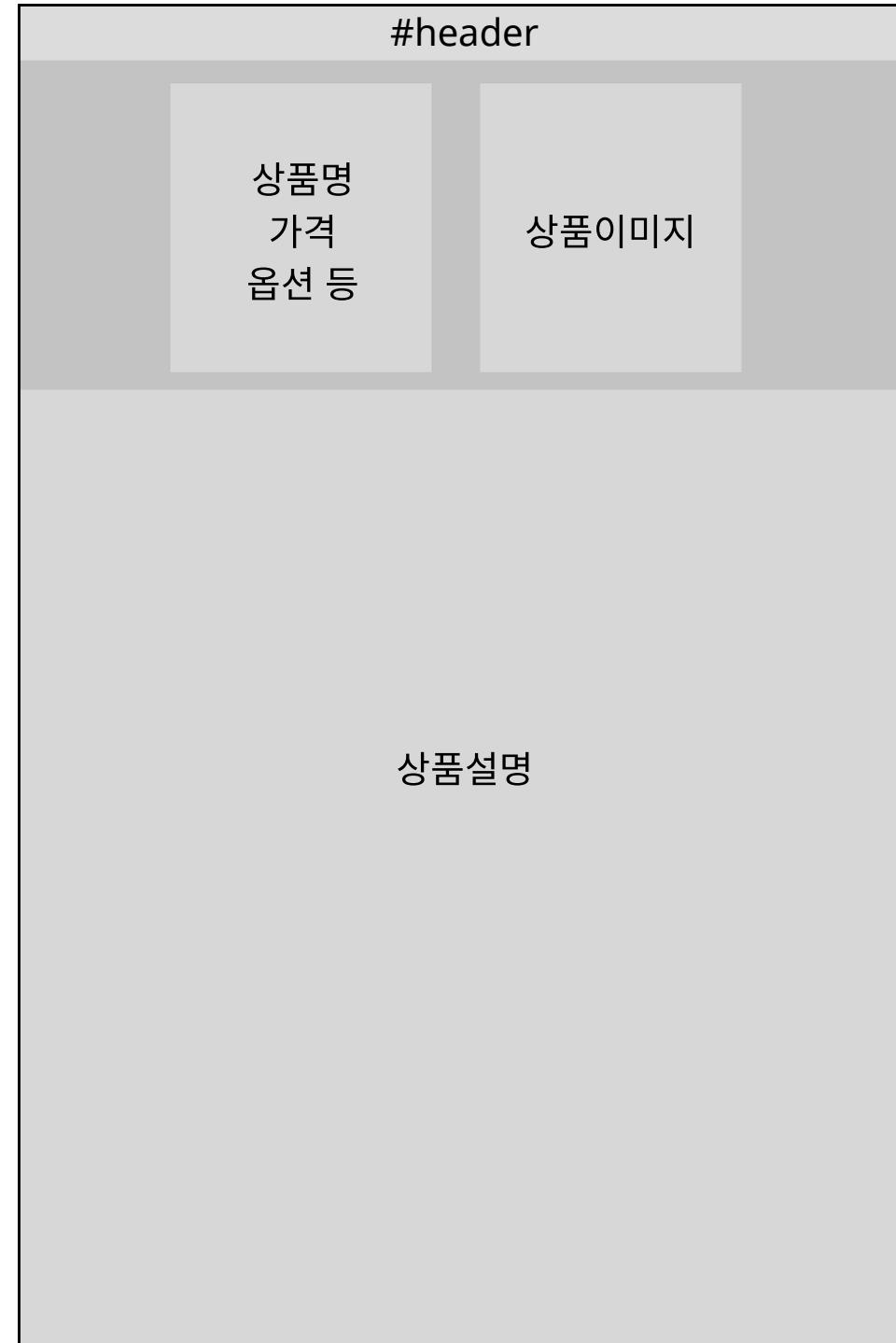
User Needs

- 콘텐츠 제작에 필요한 제품을 한눈에 파악
- 스펙 및 용도 중심의 명확한 분류
- 시각적으로 보기 쉬운 구성
- 사용 목적에 맞는 추천 정보 제공

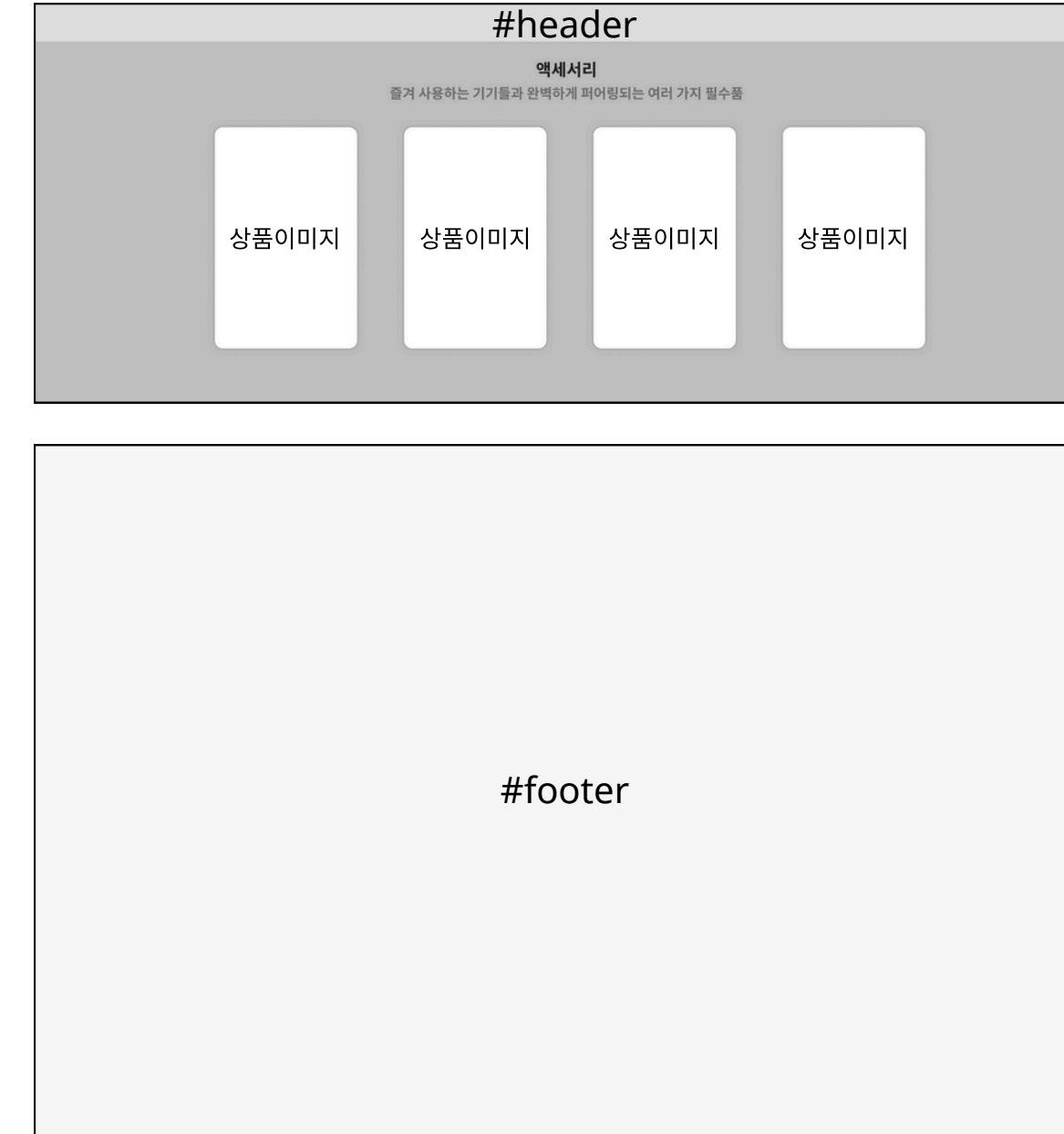
2. 기획 - 와이어 프레임 (Main)



2. 기획 - 와이어 프레임 (sub, sub_Detail)



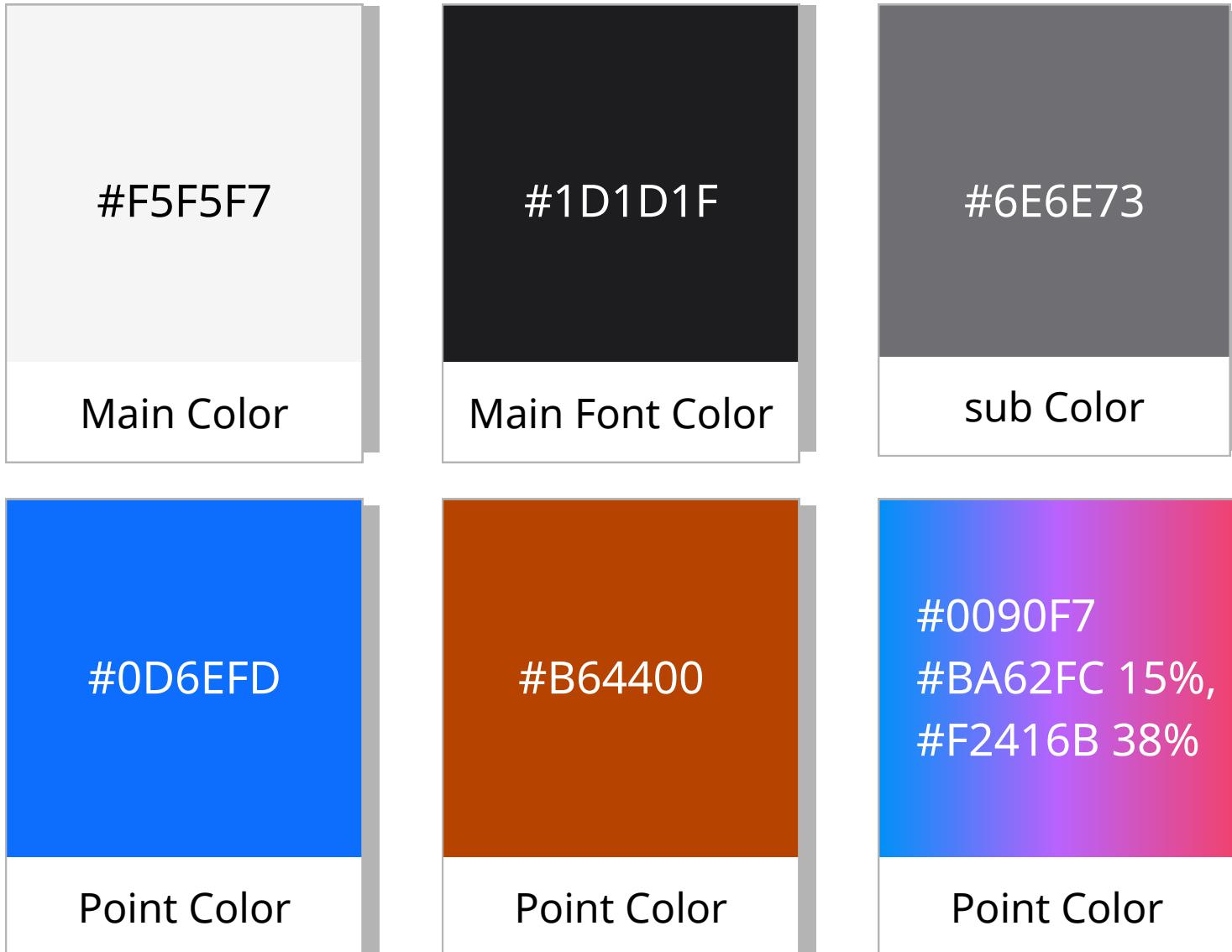
<sub>



<sub_Detail>

2. 기획 - STYLE SYSTEM

- color system



- font system

기본 글꼴 = **Inter**로 사용하였습니다.

슬라이드 제목 폰트 입니다. h2 50px

제목 폰트 입니다. h3 30px

서브 제목 폰트 입니다. p 25px

기본 내용 폰트입니다. 15px

그외 꾸미는 폰트입니다. 14px

그외 꾸미는 폰트입니다. 13px

푸터 폰트입니다. 12px

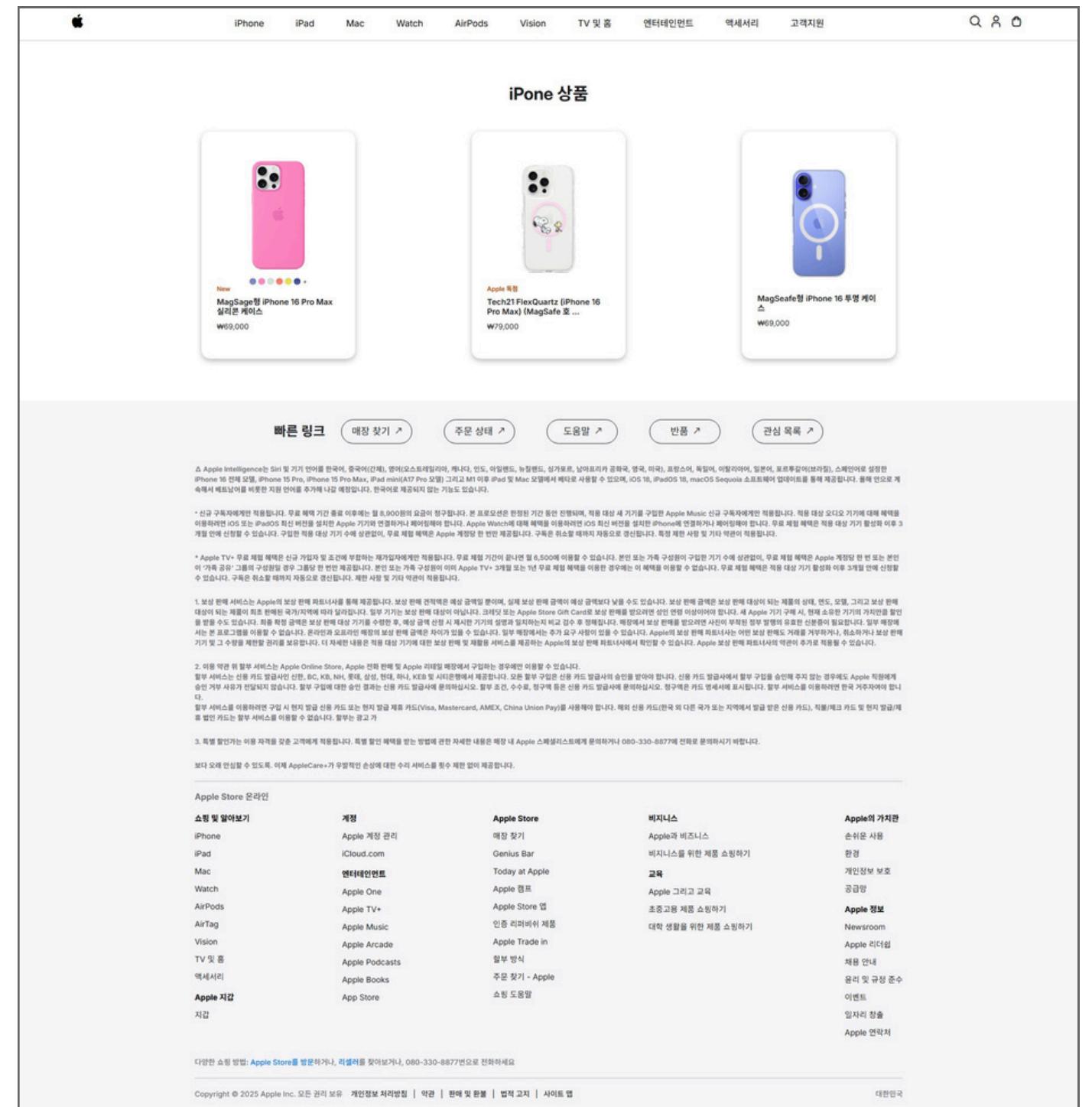
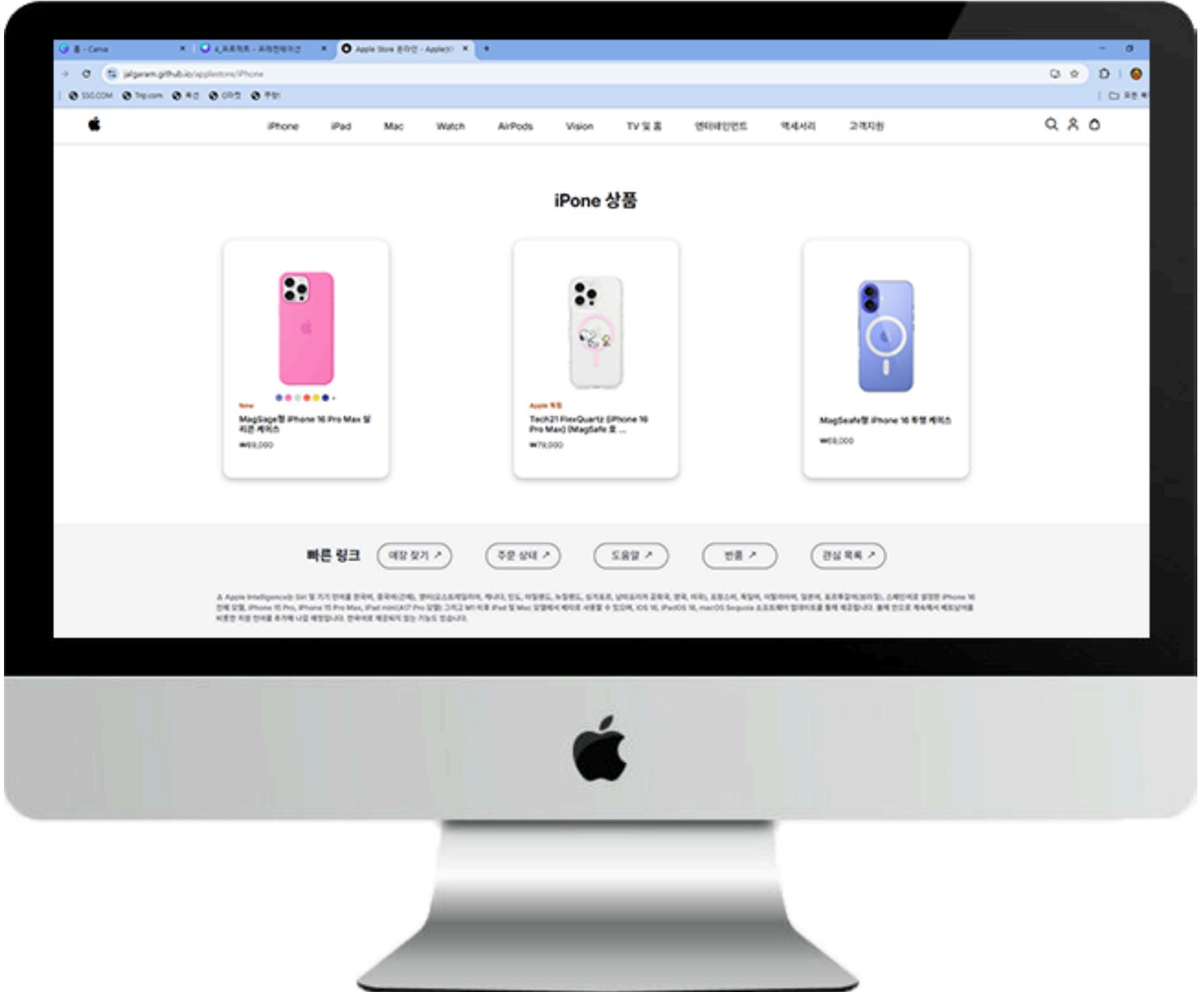
2. 기획 - 프로토타입(Main)

The main page features a large banner for the MacBook Air. Below it, there's a section for '최신제품' (New Products) featuring the iPhone 16 and iPad Air. A 'Apple 경험' (Apple Experience) section encourages users to explore products and services. At the bottom, there are promotional boxes for Apple Store online, Apple TV+, and AppleCare+.

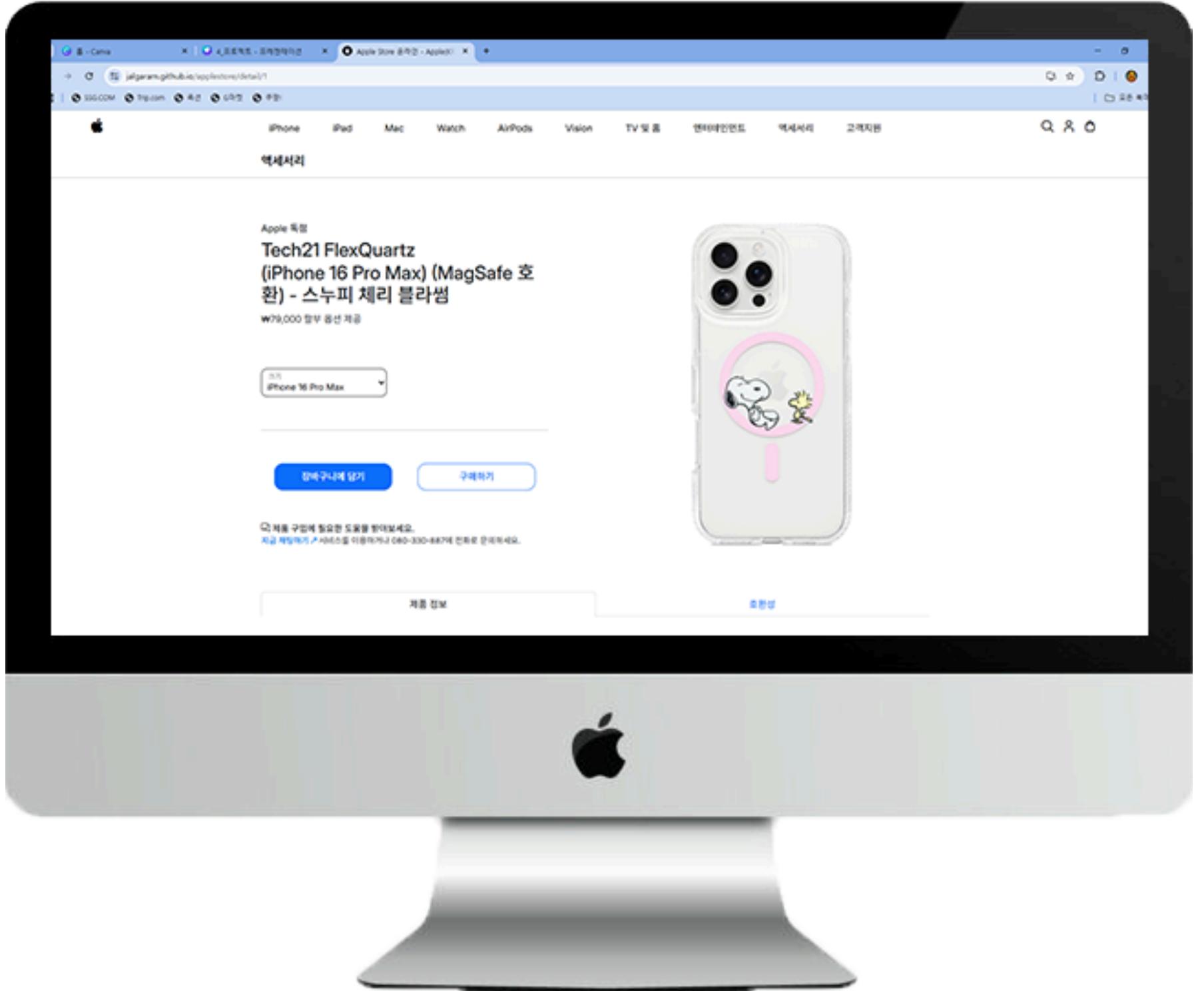
This page displays a grid of Apple accessories. It includes sections for '액세서리 상품' (Accessory Products) and '1년 보증이 제공되는 리퍼비쉬 Apple 제품을 쇼핑하세요' (Shop refurbished Apple products with a 1-year warranty). The page also features a '비즈니스' (Business) section with information about business discounts and programs.

This screenshot shows another view of the Apple accessories page, featuring a different selection of cases and covers. It follows the same layout with sections for '액세서리 상품' and '1년 보증이 제공되는 리퍼비쉬 Apple 제품을 쇼핑하세요'.

2. 기획 - 프로토타입 (sub)



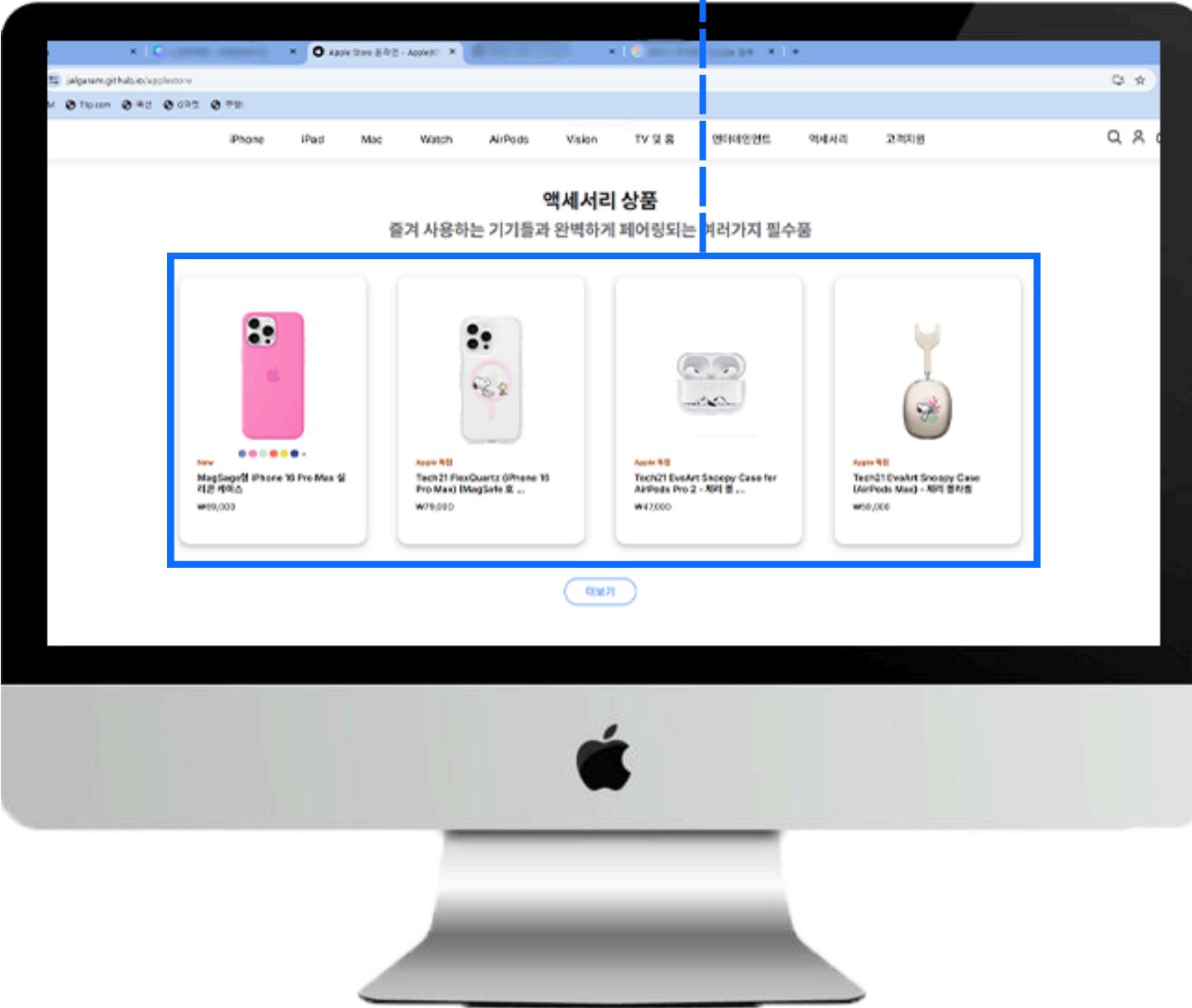
2. 기획 - 프로토타입 (sub_Detail)



This screenshot provides a detailed view of the Tech21 FlexQuartz phone case product page. It includes sections such as:

- 개요**: A general description of the product, mentioning its shock-absorbing capabilities and MagSafe compatibility.
- 하이라이트**: Key features of the product.
- 제품 구성**: Product configuration options.
- 상품정보**: Product information, including manufacturer (TECH21), country of origin (South Korea), contact information, and shipping details.
- 제품 사양**: Technical specifications: 2025년 3월 출시, 흰색 하드 케이스, TPU 및 나일론 재질, 1.4cm 높이, 17cm 길이, 8.5cm 가로, 46g 무게.
- 제품 번호**: UPC/EAN 번호 (5056586746981).
- 제조업체 정보**: Manufacturer information, noting that the manufacturer is also the service provider.
- 착불는 액세서리**: Accessories compatible with the case.

3. 구현 - Main



모든 상품 정보를 데이터로 분리하였고,
map()을 사용해 각 상품의 이미지·색상·이름·가격 등
을 반복적으로 렌더링 하여 구현하였습니다.

```
function Product1(props) {  
  
  return (  
    <Col xs={1} className="contentBox">  
      <Link to={`/detail/${props.contents.id}`}>  
        <div className="contentImgBox">  
          <img src={props.contents.img} alt="액세서리" />  
          {props.contents.imgColor && (  
            <img src={props.contents.imgColor} alt="색상" />  
          )}  
        </div>  
  
        <div className="contentfont">  
          <h6>{props.contents.smallTitle}</h6>  
          <h4>{props.contents.mainTitle.length > 50  
            ? props.contents.mainTitle.slice(0, 50) + ' ...'  
            : props.contents.mainTitle}</h4>  
          <p>₩{props.contents.price.toLocaleString()}</p>  
        </div>  
      </Link>  
    </Col>  
  )  
  
  export default Product1;
```

3. 구현 - Main

액세서리 상품
즐겨 사용하는 기기들과 완벽하게 페어링되는 여러가지 필수품

New MagSafe형 iPhone 16 Pro Max 실리콘 케이스 ₩69,000	Apple 독점 Tech21 FlexQuartz (iPhone 16 Pro Max) (MagSafe 포함) ₩79,000	Apple 독점 Tech21 EvoArt Snoopy Case for AirPods Pro 2 - 체리 플라워 ₩47,000	Apple 독점 Tech21 EvoArt Snoopy Case (AirPods Max) - 체리 플라워 ₩59,000
---------------------------------------------------	------------------------------------------------------------------------	--------------------------------------------------------------------------	----------------------------------------------------------------------

더보기

Apple 독점 Herschel Burrard Organizer Tech - 스누피 체리 플라워 블랙 ₩89,000	Apple 독점 PopSockets Magsafe Grip (iPhone) 스누피 체리 플라워 ₩59,000	Apple 독점 PopSockets Magsafe Grip (iPhone) 스누피 체리 플라워 ₩59,000	Apple 독점 Satchi Magnetic Wallet Stand(MagSafe 호환) - 스누피 체리 ₩69,000
iPad(A16모델)용 Magic Keyboard Folio ₩380,000	MagSafe형 iPhone 16 투명 케이스 ₩69,000	49mm 밀레니즈 루프 (M) ₩285,000	New Beats USB-C-USV-C 우븐 케이블 (1.5m) ₩69,000

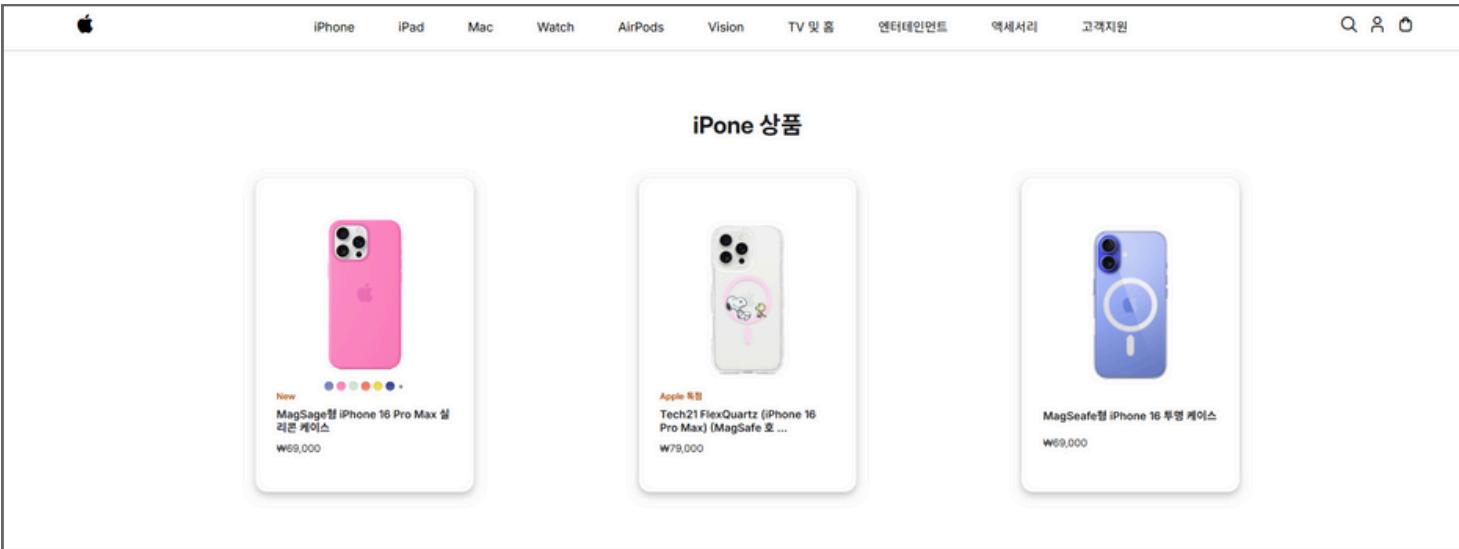
```

32 {visibleRows >= 2 && (
33   <Container>
34     <Row className='contentBigbox'>
35       {contents2.map((content) => (
36         <Product1 contents={content} key={content.id} />
37       ))}
38     </Row>
39   </Container>
40 )
41 }
42 {visibleRows >= 3 && (
43   <Container>
44     <Row className='contentBigbox'>
45       {contents3.map((content) => (
46         <Product1 contents={content} key={content.id} />
47       ))}
48     </Row>
49   </Container>
50 )
51 }
52 {visibleRows < 3 && (
53   <Button
54     variant="outline-primary"
55     style={{ padding: '7px 30px', borderRadius: '50px' }}
56     onClick={() => setVisibleRows(visibleRows + 1)}
57   >
58     더보기
59   </Button>
60 )
}

```

'더보기' 버튼 클릭 시 한 줄(4개 상품) 씩 동적으로 상품이 노출되도록 구현하였으며, 최대 3줄(12개 상품)까지 표시된 후에는 버튼이 자동으로 숨겨지도록 설정하여 하였습니다.

3. 구현 - sub



React Router를 활용해 각 카테고리별 URL 경로를 설정하였고, 헤더의 메뉴 클릭 시 해당 경로로 이동하여 카테고리에 맞는 상품 데이터를 필터링해 보여주는 기능을 구현했습니다.

```
const iPhone = () => {
  let allData = [...data, ...data2, ...data3];
  let iphoneData = allData.filter(item => item.modelType === 'iPhone' || item.modelType === 'iPhone2');

  return (
    <>
      <Header />

      <hr style={{ margin: 0 }} />

      <div className='contentWrap'>
        <div className="contentTitle">
          <h3>iPhone 제품</h3>
        </div>

        <Container>
          <Row className='contentBigbox'>
            {iphoneData.length === 0 ? (
              <p style={{margin:0, padding:0}}>해당 상품은 준비중입니다 :(</p>
            ) : (
              iphoneData.map((content) => <Product1 contents={content} key={content.id} />
            )}
          </Row>
        </Container>
      </div>
      <Footer />
    </>
  )
}

export default iPhone
```

3. 구현 - sub



아이디 입력값이 이메일 또는 전화번호 형식이 아닐 경우, 그리고 비밀번호에 공백이 포함되었거나 영문·숫자·특수문자의 조합이 아닌 경우, 혹은 동일 문자가 4회 이상 반복되는 경우 등 사용자에게 경고 메시지가 띄도록 구현했습니다.

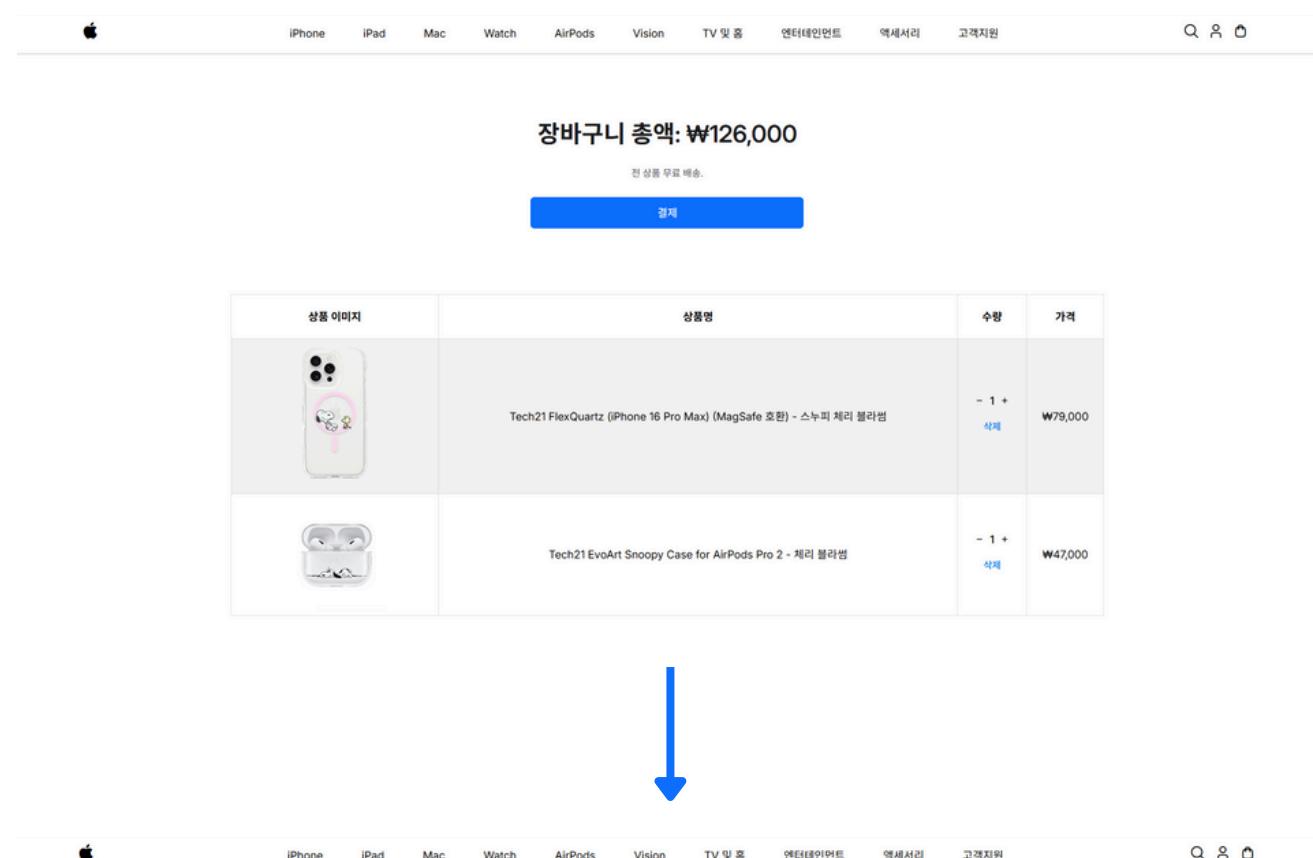
```
const Login = () => {
  const id = useRef();
  const password = useRef();

  function chkLogin(){
    var idv = id.current.value;
    var pw = password.current.value;

    var num = pw.search(/\d/g);
    var eng = pw.search(/[a-z]/ig);
    var spe = pw.search(/[^~!@#$%^&*|:;?]/gi);
    var email = `/^([\\w-]+(?:\\.(\\w-]+)*@[\\w-]+\\.\\w{1,6})\\.(\\w{2,6}(?:\\.[\\w-]{2}))$/`;
    var phone = `/^01[016789][0-9]{7,8}$/`;

    if(!email.test(idv) && !phone.test(idv)){
      alert('이메일 또는 전화번호 형식이 아닙니다.');
      return false;
    } else if(pw.length < 8 || pw.length > 20){
      alert('8자리 ~ 20자리 이내로 입력해주세요.');
      return false;
    } else if(pw.search(/\s/) != -1){
      alert('비밀번호는 공백 없이 입력해주세요.');
      return false;
    } else if((num < 0 && eng < 0) || (eng < 0 && spe < 0) || (spe < 0 && num < 0)){
      alert('영문, 숫자, 특수문자 중 2가지 이상을 혼합하여 입력해주세요.');
      return false;
    } else if(/(\w)\1\1\1/.test(pw)){
      alert('같은 문자를 4번 이상 사용할 수 없습니다.');
      return true;
    } else {
      alert('로그인 되었습니다.');
      return true;
    }
  }
}
```

3. 구현 - sub



장바구니 총액: ₩126,000

상품 이미지	상품명	수량	가격
	Tech21 FlexQuartz (iPhone 16 Pro Max) (MagSafe 호환) - 스누피 체리 블라썸	- 1 + 삭제	₩79,000
	Tech21 EvoArt Snoopy Case for AirPods Pro 2 - 체리 블라썸	- 1 + 삭제	₩47,000

장바구니 총액: ₩195,000

상품 이미지	상품명	수량	가격
	Tech21 FlexQuartz (iPhone 16 Pro Max) (MagSafe 호환) - 스누피 체리 블라썸	- 1 + 삭제	₩79,000
	Tech21 EvoArt Snoopy Case for AirPods Pro 2 - 체리 블라썸	- 1 + 삭제	₩47,000
	Beats USB-C-USV-C 우분 케이블 (1.5m)	- 1 + 삭제	₩69,000

```
cart.map((v, i) =>
  <tr key={i}>
    <td>
      <img src={v.img} alt="상품이미지"
            style={{cursor:"pointer"}}
            onClick={() => {navigate(`/detail/${v.id}`)}}
      /></td>

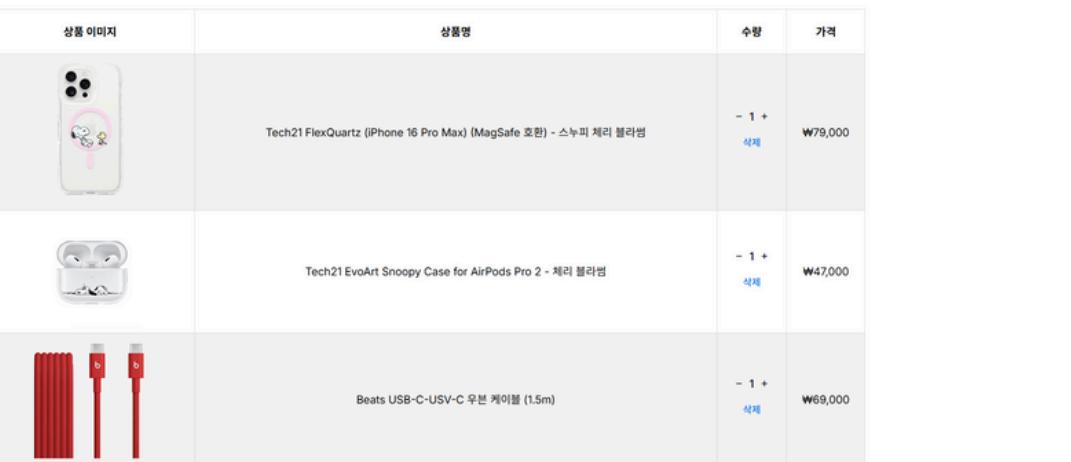
    <td className='cartTitle'
        style={{cursor:'pointer'}}
        onClick={() => {navigate(`/detail/${v.id}`)}}>{v.mainTitle}</td>

    <td className='cartCount'>
      <button onClick={() => {
        if(cart[i].amount <= 1){
          alert('최소 주문 수량은 1개입니다.')
        } else {
          dispatch(minusCount(i));
        }
      }}>
        <i className="fi fi-sr-minus-small"></i>
      </button>

      <span> {v.amount} </span>

      <button onClick={() => dispatch(addCount(i))}>
        <i className="fi fi-sr-plus-small"></i>
      </button>
    </td>

    <p className='cartDelt' onClick={() => dispatch(removeItem(i))}>
      삭제
    </p>
  </td>
</tr>
)
```

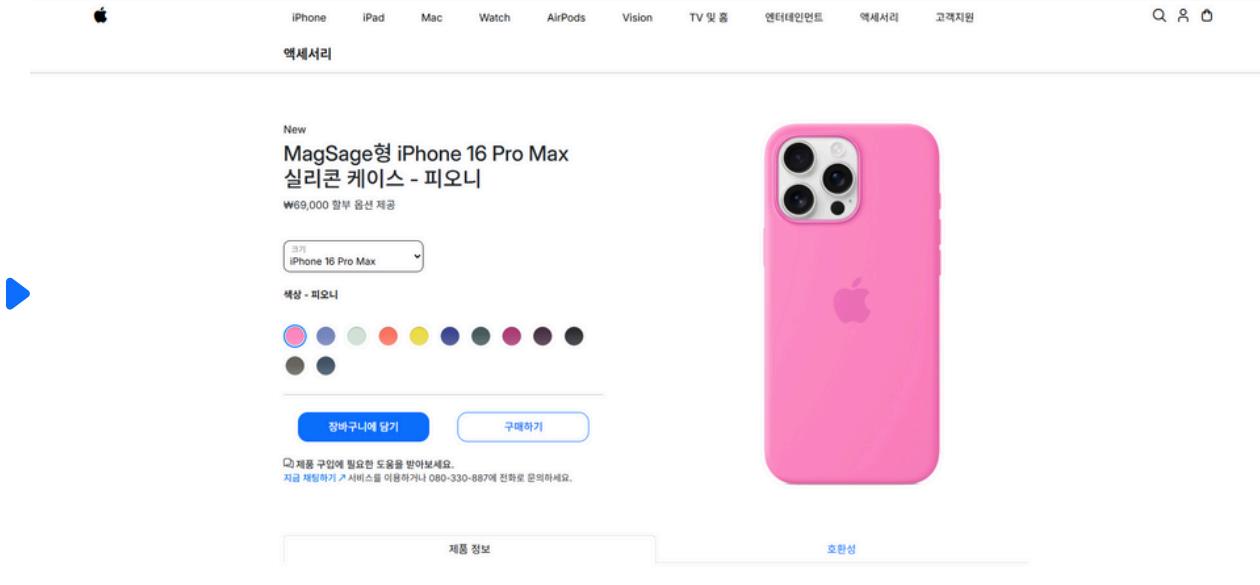
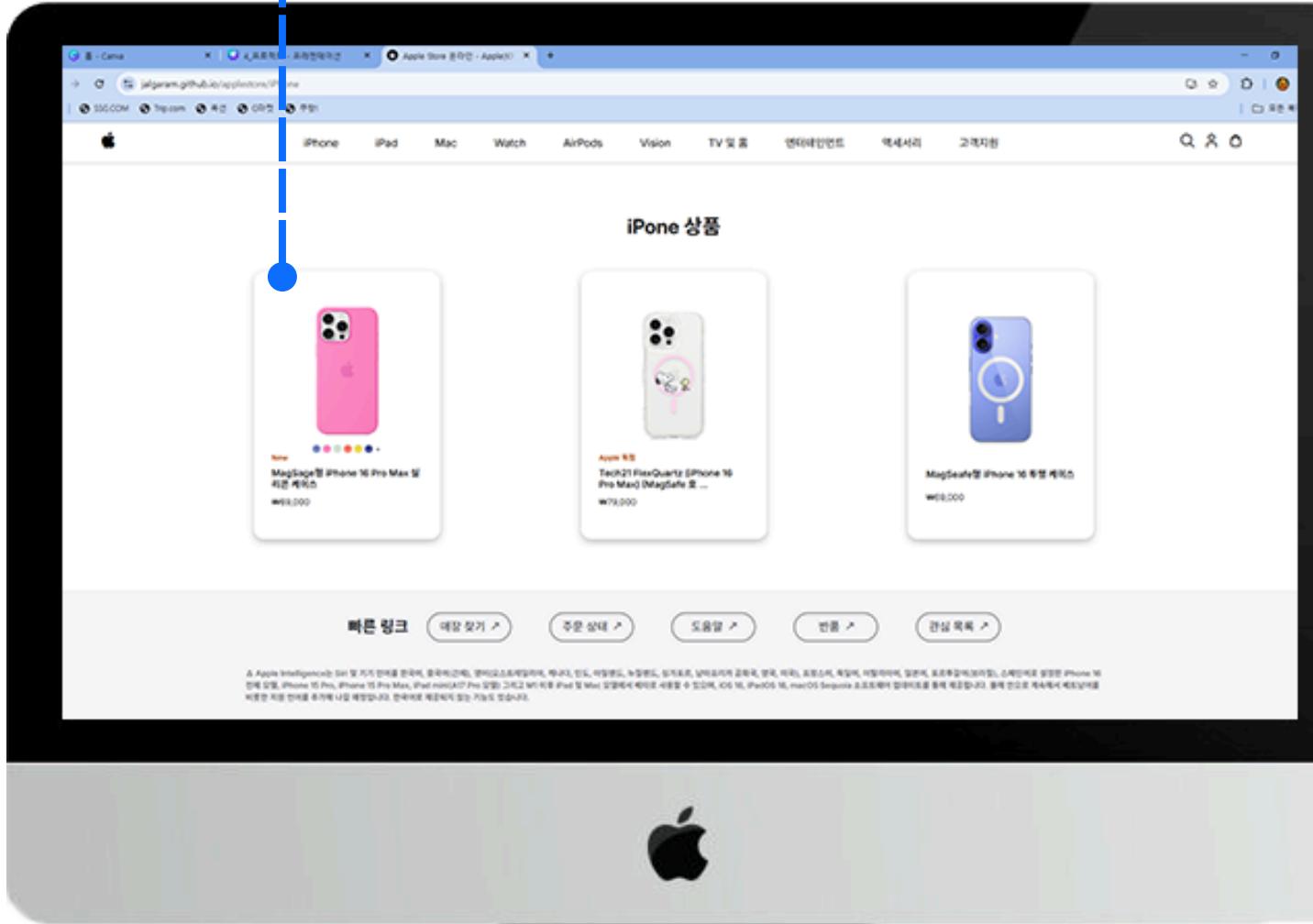


장바구니 총액: ₩195,000

상품 이미지	상품명	수량	가격
	Tech21 FlexQuartz (iPhone 16 Pro Max) (MagSafe 호환) - 스누피 체리 블라썸	- 1 + 삭제	₩79,000
	Tech21 EvoArt Snoopy Case for AirPods Pro 2 - 체리 블라썸	- 1 + 삭제	₩47,000
	Beats USB-C-USV-C 우분 케이블 (1.5m)	- 1 + 삭제	₩69,000

+ , - 버튼 클릭 시 수량이 1씩 증감되도록 구현하였고, ‘삭제’ 버튼을 클릭하면 해당 상품이 장바구니에서 제거되도록 구성했습니다. 또한, 상품 상세페이지에서 ‘장바구니 담기’ 버튼을 클릭하면 해당 상품이 장바구니에 추가되며, 총액도 실시간으로 반영되도록 구현했습니다.

3. 구현 - sub_Detail



사용자가 상품을 클릭하면 해당 상품의 상세 페이지로 이동하게 구현했습니다.

```
const DetailContent = ({ product }) => {
  const dispatch = useDispatch();
  const navigate = useNavigate();

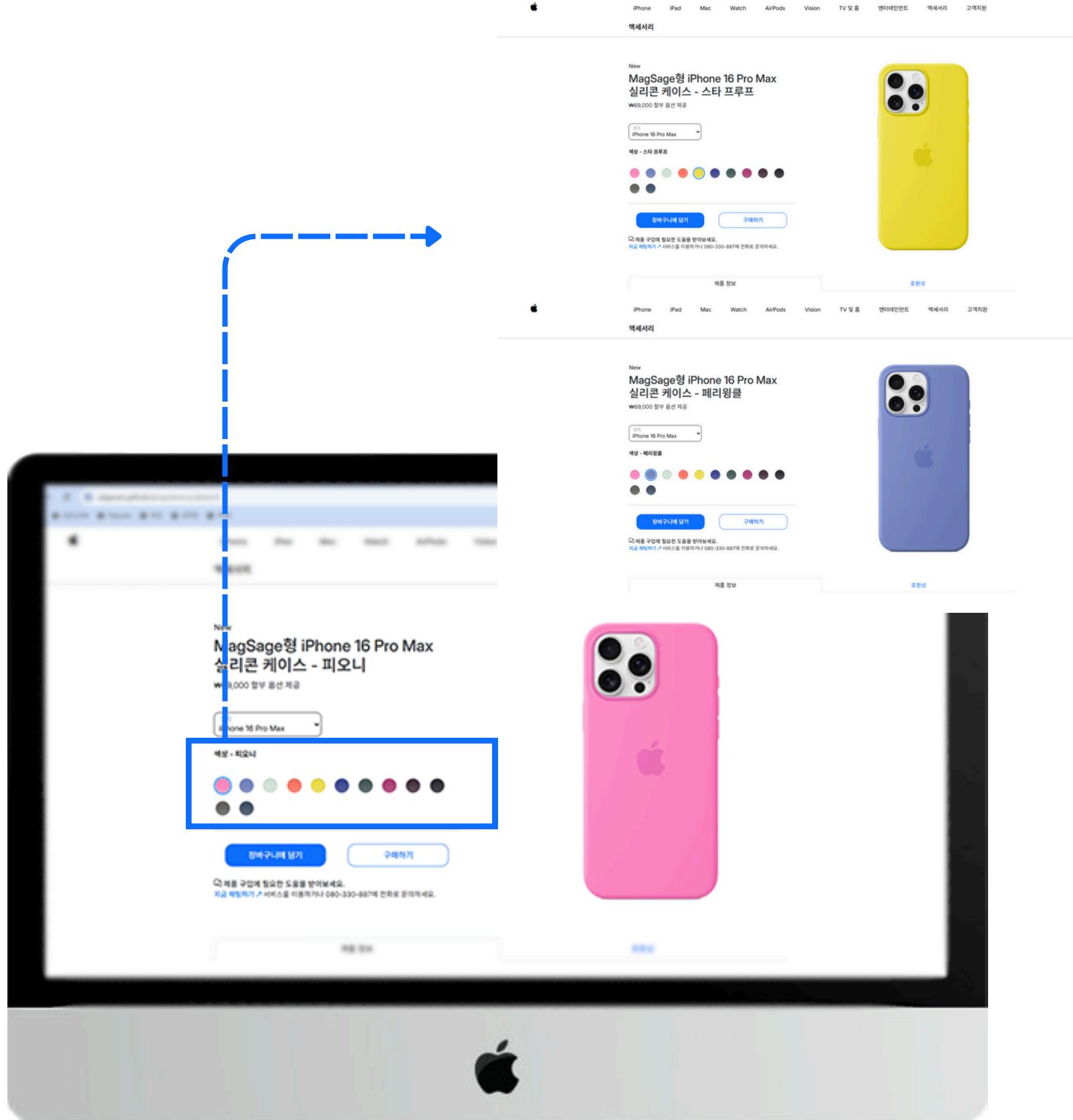
  const [selectedIndex, setSelectedIndex] = useState(0);
  const [hovername, setHovername] = useState(null);

  if (!product) return null

  return (
    <div className='detailContent_wrap'>
      <div className="detailheader">
        <h3>액세서리</h3>
      </div>
      <hr />

      <Container className="detailBigBox">
        <Row className="detailBox1">
          <Col xs={6} className="detailFontBox">
            <div className="detailfontWrap">
              <h2 style={{ whiteSpace: 'pre-line' }}>{product.title} {product.colors?.[selectedIndex]?.name || ' - ${product.colors[selectedIndex].name}' }</h2>
              <p>₩{product.price.toLocaleString()} <span onClick={(e) => {
                e.preventDefault();
                window.open(
                  process.env.PUBLIC_URL + '/img/Untitled.png',
                  '할부옵션제공안내',
                  'width=580,height=882');
              }}>할부 옵션 제공</span></p>
            </div>
          </Col>
        </Row>
      </Container>
    </div>
  )
}
```

3. 구현 - sub_Detail



```
{product.colors && (
  <>
    <h4>색상 - <span>{product.colors[hovername ?? selectedIndex].name}</span></h4>

    <div className="color">
      {product.colors.map((color, index) => (
        <img
          key={index}
          src={color.src}
          alt={color.name}
          className={selectedIndex === index ? "active" : ""}
          onClick={() => setSelectedIndex(index)}
          onMouseEnter={() => setHovername(index)}
          onMouseLeave={() => setHovername(null)}
        />
      )));
    </div>
  )}
```

여러 색상 옵션이 있는 상품의 경우, 사용자가 색상을 선택하면 해당 색상에 맞는 상품 이미지와 색상명이 동적으로 변경되어 표시되며, 현재 선택된 색상은 파란색 원형 표시로 시각적으로 구분되도록 구현했습니다.

4. 마무리 - 평가 및 후기

React 수업을 들을 때는 내가 제대로 이해하고 있는지 확신이 들지 않았지만, 실제로 프로젝트를 진행해 보니 HTML보다 React가 훨씬 더 편하다는 것을 느꼈습니다.

특히 JavaScript 파일을 연동해 데이터를 따로 관리하고, map()을 활용해 반복 렌더링 하는 방식이 매우 효율적이었고, 각 섹션을 컴포넌트 단위로 나누어 관리할 수 있어 작업이 훨씬 수월했습니다.
또한 React-Bootstrap을 함께 사용하면서, 이전에 HTML과 CSS(SCSS) 만으로 작업했던 프로젝트보다 개발 속도도 훨씬 빨라졌습니다.

이번 프로젝트를 통해 React의 구조적 장점과 실무에서의 편리함을 확실히 체감할 수 있었고, 이전에 만들었던 프로젝트들도 React를 활용해 리팩토링하거나 보완해 보고 싶다는 생각이 들었습니다.

Thank You

✉ <https://jalgaram.github.io/applestore/>
