

Project Proposal

Presented by Young A Kim

Contents

001 Project Research

Market Research

Brain storming

Persona

002 Strategy

Wireframe

UI UX

003 website Building

Code / Implement

004 Review

Retrospective



Timeline

1W

2W

3W



Project concept

design

Layout & prototype

building

final





001

Project Research

001 Project Research

Market Research

개성과 패션

MZ(밀레니얼+Z) 세대들은 트렌드에 민감하고 자신을 표현하며 새로운 스타일을 연출하는 것에 수요가 높기 때문에 재미와 감각을 더한 콘텐츠들이 눈과 지갑을 사로 잡고 있습니다.

하나의 트렌드 ‘백꾸’

일명 ‘백 꾸미기’라고 불리는 가방 꾸미기이다.
요즘 세대의 트렌드로 떠오르면서 가방을 꾸밀 수 있는 다양한 아이템들을 찾는 소비자가 늘고 있는 추세다. 패션 브랜드들도 키링 아이템을 속속 출시하는 추세입니다.

001 Project Research

Brain storming

트렌드

인테리어

립스틱 효과

팝업 스토어

창작과 디자인

셀럽

귀여운 소품

자기표현

빈티지

감각적인

001 Project Research

Persona 01



Name : 김지우

Age : 25

Location : 경기도 수원시

Work : 대학생

Character : 호기심이 많고 적극적인 성격

Attitude

일상의에서의 자신과 타인, 그리고 만나는 모든 것들에 대해 진심과 사랑으로 대합니다.
사람에게 관심을 주고 받는 것에 행복을 느낍니다. 그래서 보이는 것에 관심이 아주 많아요.
나의 보이는 모든 것이 매력적인 사람처럼 비춰졌으면 좋겠습니다.

intro

스타일링에 관심이 많아서 매일 대학에 갈 때 어떻게 꾸밀지를 고민하는 시간을 가지고 있어요.
자신만의 스타일을 찾는 데 열정적이고, 스타일링 아이템을 활용하여 매일 새로운 자신의 모습으로 하루를 보낸답니다.
꾸미는 것을 좋아하는 김지우는 자신만의 개성을 표현하기 위해 노력하고 있어요.



001 Project Research

Persona 02



Name : 손동표

Age : 23

Location : 경기도 광교시

Work : 유튜브 크리에이터

Character : 긍정적인 성격

Attitude

한가지에 끌히면 그 주제만 깊이 있게 정보를 수집 하며 알아갑니다. 크리에이터로서 자신의 경험 뿐만 아니라 구독자들의 의견과 생각도 존중하며, 이를 통해 다양한 콘텐츠를 만들어 나가려고 해요. 많은 사람들과 긍정적인 영향을 나누고 싶어합니다.

intro

나의 채널에서 다양한 콘텐츠로 사람들과 소통하고 싶어요. 뻔하지 않은 주제로 가득 채워서 구독자를 늘리고 싶어요. 연령대별 관심사가 무엇인지 파악하며 공부하고 있답니다. 소품을 구매하여 리뷰하는 방식으로 콘텐츠를 제작 해볼 계획 이에요.





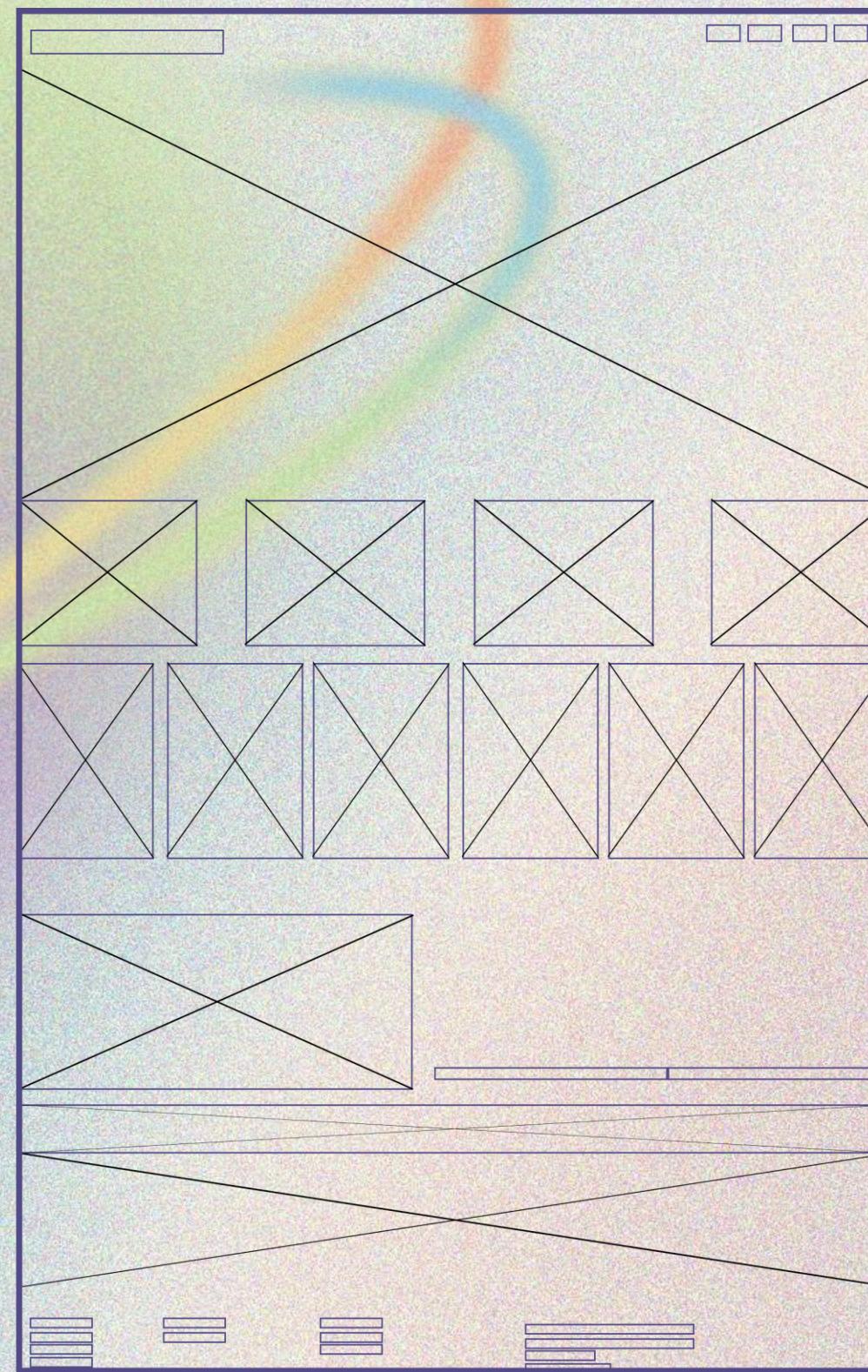
002

Strategy



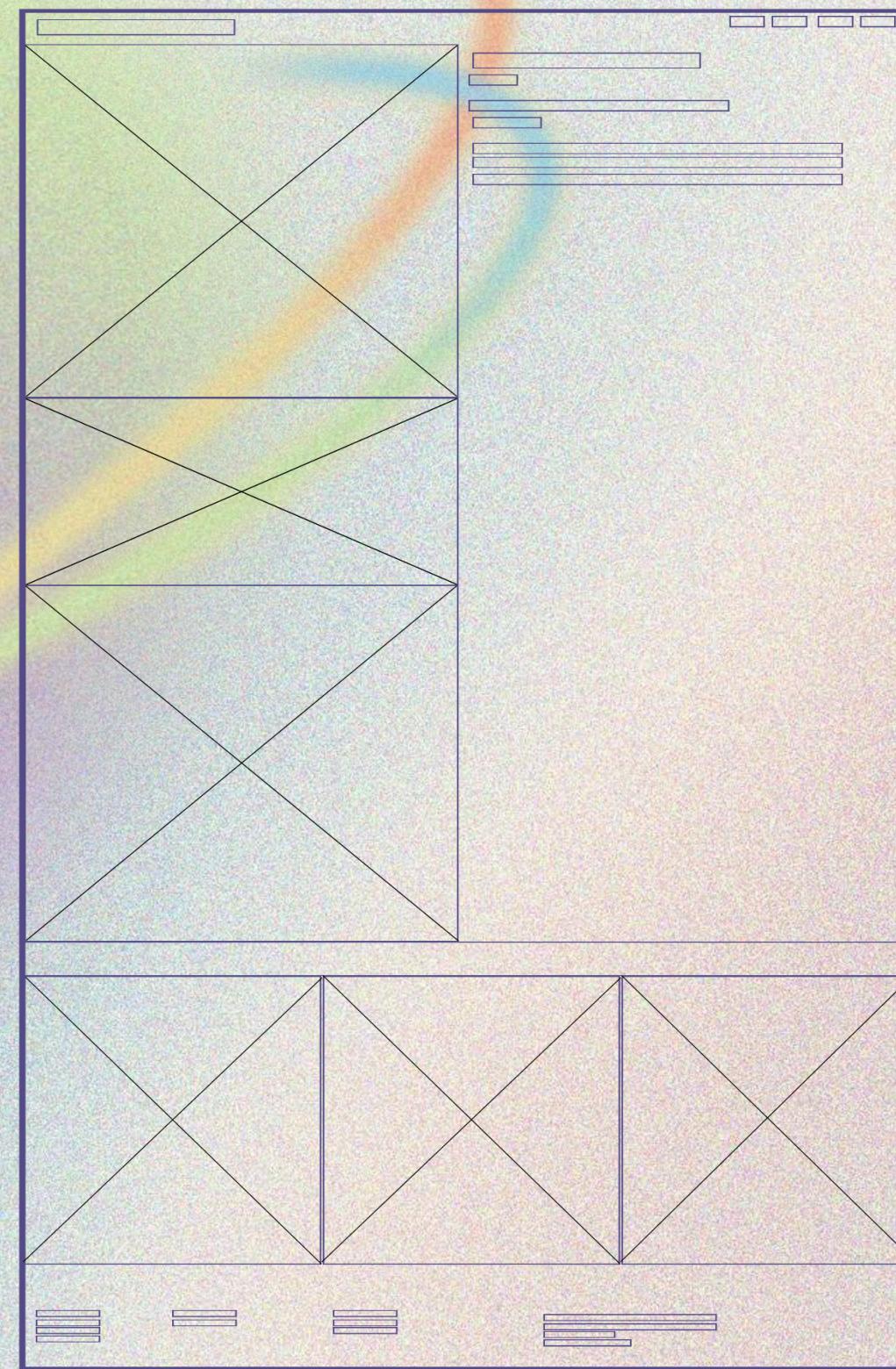
002 Strategy

Wireframe Home



002 Strategy

Wireframe detail



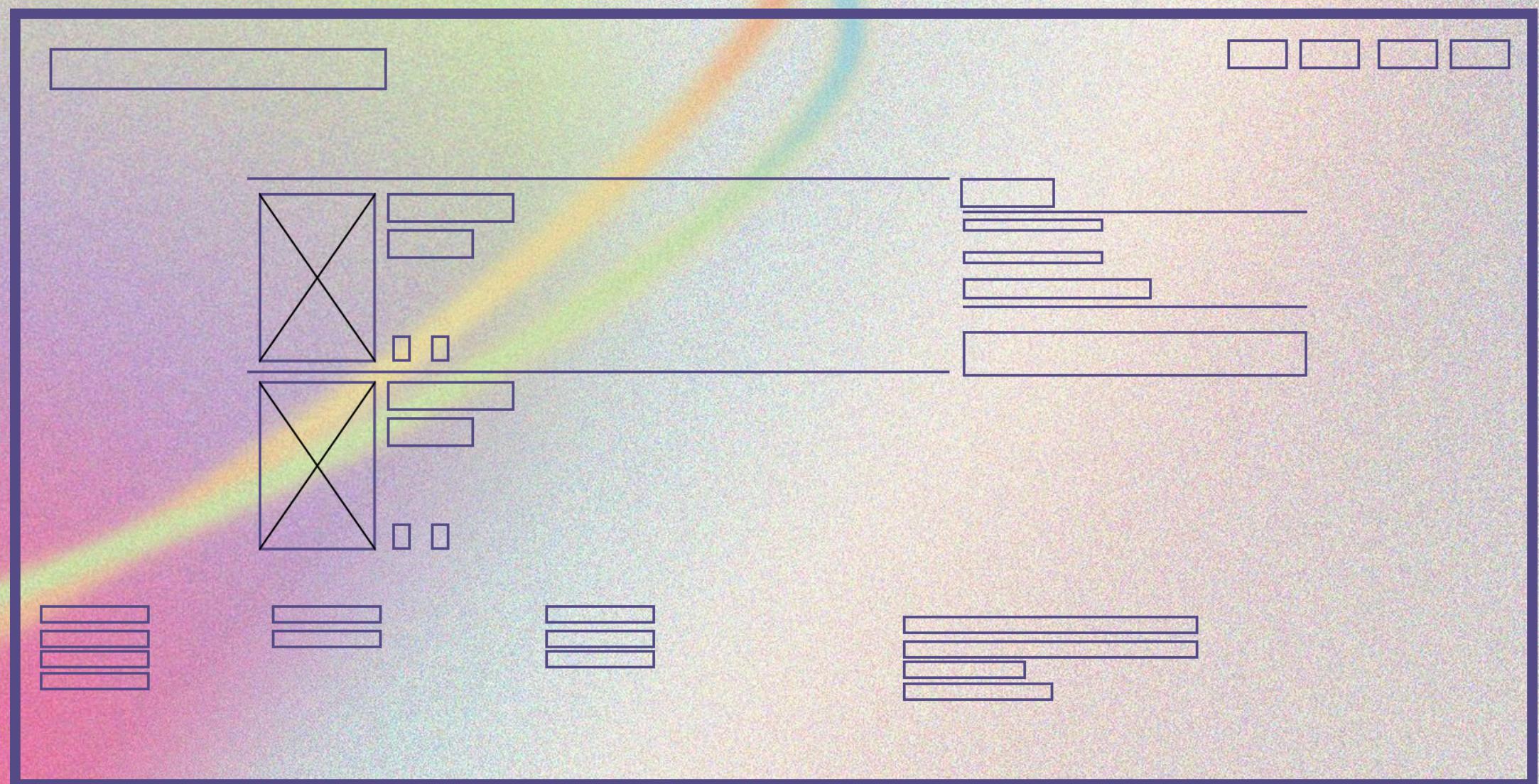
002 Strategy

Wireframe Login



002 Strategy

Wireframe Cart



002 Strategy

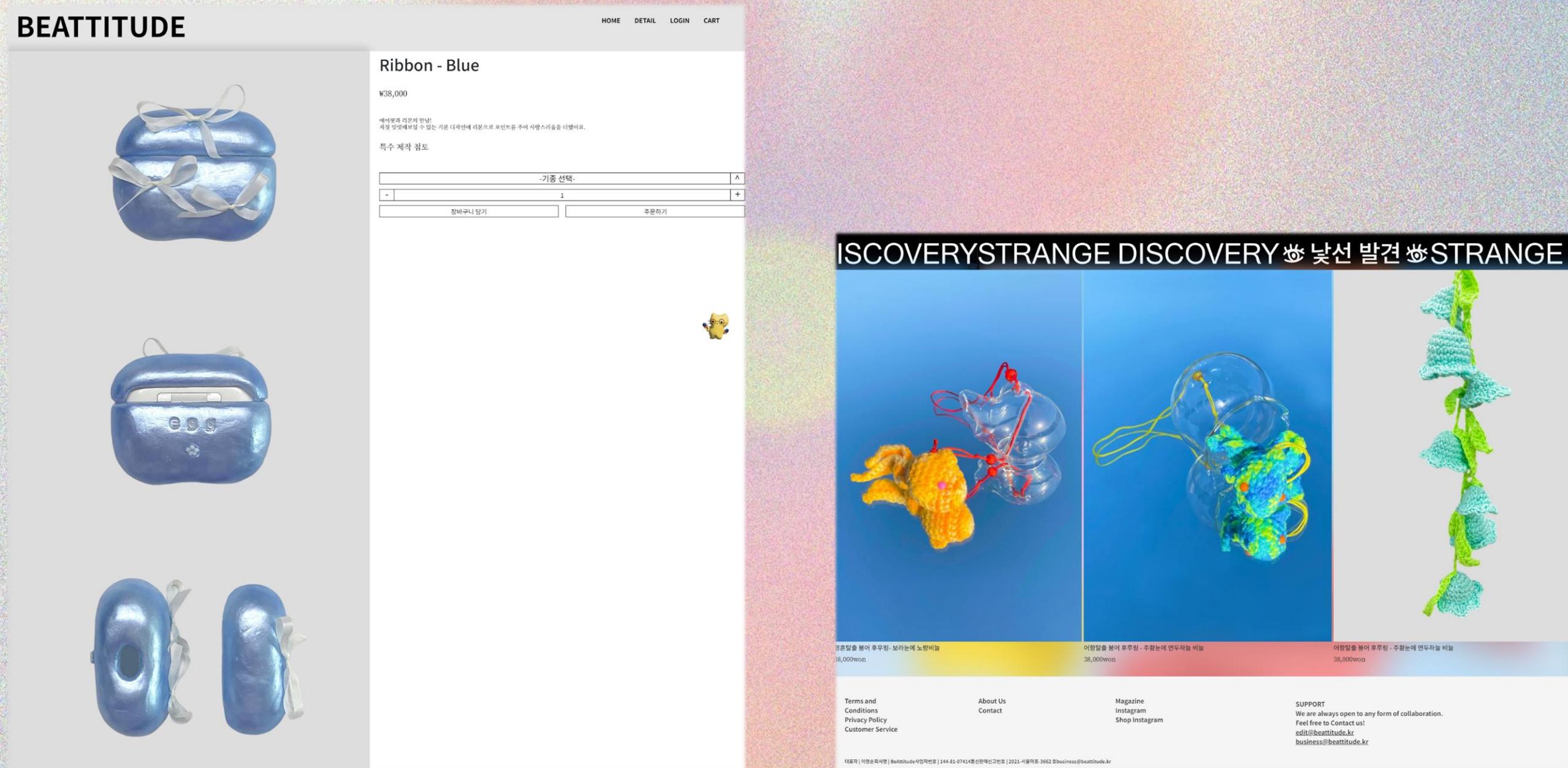
UI UX Home

The home page of the Beatitude website features a vibrant, colorful header with a yellow-to-pink gradient. The title "BEATTITUDE" is at the top left, and a navigation bar with links to HOME, DETAIL, LOGIN, and CART is at the top right. The main visual is a whimsical illustration of a white coffee cup with a saucer, surrounded by red and pink heart-shaped confetti, a green star-shaped pillow with the text "TAKE A NAP", a pink speech bubble with "get up late or sleep", and a clear cube containing a small cake. Below the illustration, the text "TUDE-DAY NEW BRAND" and "baked.studio & EZR & New Dungeon & dodstudio" is displayed. A grid of product cards below shows various items: a phone case, an earpod case, a smart strap, a keyring, and six different phone cases with unique designs like a tea spoon set and a flower pattern.

This page has a pink-to-white gradient background. At the top, there's a section titled "작가의 이야기" (Artist's Story) featuring three rows of small, colorful illustrations related to the brand's theme. To the right, a text box discusses the artist's philosophy: "도너리 작가는 스스로 귀여움을 오랫동안 헤쳐온는 악후라고 고백해요. 귀여운 것은 절 때면 다양한 아이디어가 생각나고, 이를 적용해 헬트에 손바느질한 귀여운 친구들을 세상에 선보이죠. 이들이 사람들에게 따뜻한 위로를 전하고, 무한한 긍정을 채워줍니다. '직장하지 마세요!'라고 말하면서도, 매달 새로운 귀여움을 사람들과 공유하는 약속을 지키기 위해 체육, 연달 관리에 힘쓰는 도너리 작가, 작은 행복이 오래가는 평온한 삶을 꿈꾸는 그의 이야기". Below this is a large banner with the text "VERYSTRANGE DISCOVERY 낯선 발견 STRANGE DISCO". At the bottom, there's a call-to-action: "°.° beattitude Accessories Shop °.°" and a message: "비애디튜드의 다양한 상품을 통해 창조의 기쁨을 함께 나눠보세요." The footer contains links for Terms and Conditions, Privacy Policy, Customer Service, About Us, Contact, Magazine, Instagram, and SUPPORT.

002 Strategy

UI UX detail



002 Strategy

UI UX Login

BEATTITUDE

HOME DETAIL LOGIN CART

LOG IN

아이디

비밀번호

Log In

Terms and Conditions
Privacy Policy
Customer Service

About Us
Contact

Magazine
Instagram
Shop Instagram

SUPPORT
We are always open to any form of collaboration.
Feel free to Contact us!
edit@beattitude.kr
business@beattitude.kr



대표자 | 이영순회사명 | BeAttitude사업자번호 | 144-81-07414통신판매신고번호 | 2021-서울마포-3662 호business@beattitude.kr

002 Strategy

UI UX Cart

HOME DETAIL LOGIN CART

BEATTITUDE

CART(3)

휴대폰 케이스 ₩23,000	X	결제 정보
Ribbon omelet rice key ring ₩18,000	X	총 주문금액 : ₩72,800 배송비 : ₩3,000 결제금액 : ₩75,800
Dekidoki ₩31,800	X	CHECKOUT

A small yellow cartoon character is positioned on the right side of the page.

TERMS AND CONDITIONS

Terms and Conditions
Privacy Policy
Customer Service

ABOUT US

About Us
Contact

SUPPORT

We are always open to any form of collaboration.
Feel free to Contact us!
edit@beattitude.kr
business@beattitude.kr

대표자 | 이영순회사명 | BeAttitude사업자번호 | 144-81-07414통신판매신고번호 | 2021-서울마포-3662 호 business@beattitude.kr



003

Website Building

003 Website Building

Code Implement1



네비게이션 바의 텍스트를 클릭할 때 각 페이지로 이동하도록 구현했습니다.
4개의 link에 navigate를 사용하여 페이지 이동을 구현하고
Route path element으로 각 이동할 페이지를 지정하고
Phone 객체를 props로 전달합니다.
사용자가 특정 경로로 이동하면 해당 페이지가 보여집니다.

```
import { Col, Nav, Navbar, Row } from 'react-bootstrap'; ...
```

```
function App() {
  const navigate = useNavigate();
  const [phone, setPhone] = useState(data)
```

```
  return (
    <div className="App">
      <Navbar collapseOnSelect expand="md" className="bg-body-transparent">
```

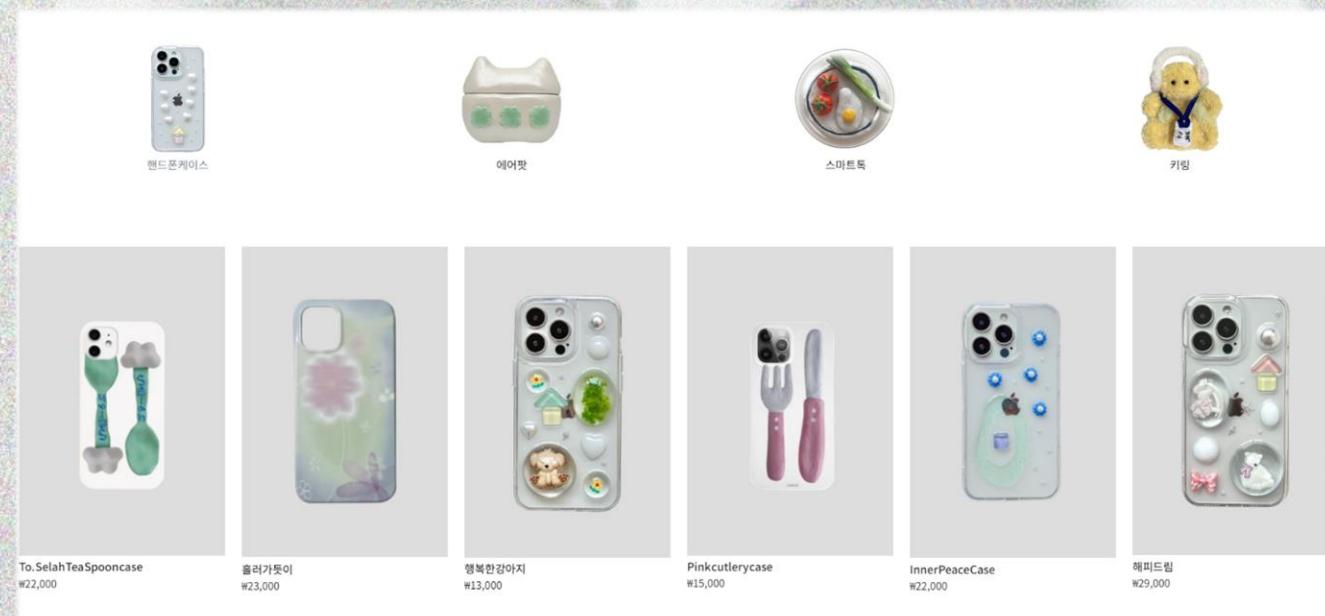
```
        <Navbar.Brand onClick={() => {
          navigate('/')
        }}>BEATTITUDE</Navbar.Brand>
        <Navbar.Toggle aria-controls="responsive-navbar-nav" />
        <Navbar.Collapse id="responsive-navbar-nav">
          <Nav className="me-auto">
            <Nav.Link onClick={() => {
              navigate('/')
            }}>HOME</Nav.Link>
            <Nav.Link onClick={() => {
              navigate('/detail/0')
            }}>DETAIL</Nav.Link>
            <Nav.Link onClick={() => {
              navigate('/login')
            }}>LOGIN</Nav.Link>
            <Nav.Link onClick={() => {
              navigate('/cart')
            }}>CART</Nav.Link>
          </Nav>
```

```
      <Routes>
```

```
        <Route path="/" element={<Main phone={phone} setPhone={setPhone} />} />
        <Route path="/detail/:id" element={<Detail phone={phone} />} />
        <Route path='/cart' element={<Cart />} />
        <Route path='/login' element={<Login />} />
      </Routes>
```

003 Website Building

Code Implement2



```
const [isActive, setIsActive] = useState([true, false, false, false])  
  
const navigate = useNavigate();  
const filterPhone = (cate) => {  
  let copy = [...origin].filter(v => v.category === cate)  
  setPhone(copy)  
}  
  
useEffect(() => {  
  filterPhone("phone")  
}, [])  
  
<Row>  
  
  <Col md={3} onClick={() => {  
    setIsActive([true, false, false, false])  
    filterPhone("phone")  
}}>  
    <button className={isActive[0] ? 'active' : ""}>  
      <img src={process.env.PUBLIC_URL + "/img/C01img1.png"} alt="img1" width="100%" />  
      <p>핸드폰케이스</p>  
    </button>  
  </Col>
```

필터링 기능을 구현했습니다. Button을 클릭하면 각 카테고리의 전달받은 phone의 데이터가 보입니다. Is Active를 true, false, ..., ..., 으로 설정하여 첫번째 버튼이 활성화된 것처럼 보이게 하였고 use Effect로 부작용 없이 데이터를 가져오게 하였습니다.

003 Website Building

Code Implement3



```
<Carousel>
  <Carousel.Item>
    <Carousel.Caption>
      <div className='textbox'>
        <h3>TUDE-DAY NEW BRAND</h3>
        <p>baked.studio & EZR & New Dungeon & dodstudio</p>
      </div>
    </Carousel.Caption>
    <img src={process.env.PUBLIC_URL +"/img/slide01.jpg"}width="100%" alt="" />
  </Carousel.Item>
```

Bootstrap의 carousel 컴포넌트를 사용하여 이미지 슬라이더를 구현했습니다.

Carousel Item으로 각 슬라이드 이미지가 나타나며 Carousel Caption 으로 각 이미지의 텍스트를 작성 하였습니다. 이미지 경로는 PUBLIC_URL을 사용하여 상대 경로로 지정했습니다.

003 Website Building

Code Implement4



VERYSTRANGE DISCOVERY 낯선 발견 STRANGE DISCO



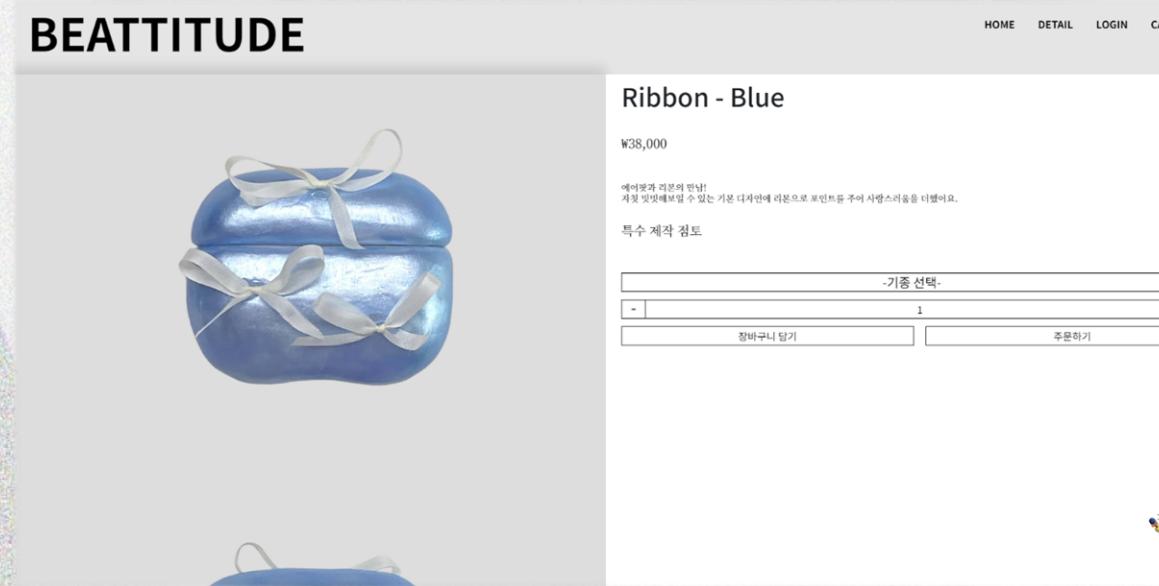
STRANGE DISCOVERY STRANGE DISCOVERY 낯선 발견

```
animation: marquee 30s linear infinite alternate;
@keyframes marquee {
  0%{
    transform: translateX(0);
  }
  100%{
    transform: translateX(-5755px);
  }
}
```

검정색 배너의 animation 기능을 구현하였습니다. 0%일 때 x축이 0이고 100%일 때 x축을 -5755px로 적용하여 infinite로 반복되게 설정하였습니다.

003 Website Building

Code Implement5



```
import { useNavigate, useParams } from 'react-router-dom';
import { addItem } from '../store';
import { useDispatch } from 'react-redux';
const dispatch = useDispatch();
const navigate = useNavigate();
const handleAddToCart = () => {
  dispatch(addItem({
    id: phone[id].id,
    imgUrl: phone[id].imgUrl,
    item: phone[id].title,
    price: phone[id].price,
    amount: 1
  }));
  navigate('/cart');
};
<Button className='shop-ba' onClick={() => handleAddToCart()}>장바구니 담기</Button>
```

장바구니에 상품을 추가하는 기능을 구현하였습니다(handle Add To Cart 함수에서 add item을 dispatch 함). Add item으로 각 상품의 정보 객체를 전달하고 navigate로 장바구니 페이지로 이동시킵니다.

003 Website Building

Code Implement6



The screenshot shows the Beatitude website's cart page. The cart contains three items:

- Plastic rice ball: ₩23,000
- Ribbon omelet rice key ring: ₩18,000
- Rice ball rice: ₩34,800

The total price is ₩75,800. There is a "CHECKOUT" button at the bottom.

```
subtractCount(state, action) {
  let idx = state.findIndex((a) => { return a.id === action.payload })
  //console.log(num);
  state[idx].amount--

  if (state[idx].amount < 1) {
    alert('최소 주문 수량은 1개 입니다.');
    state[idx].amount = 1;
  }

  addCount(state, action) {
    let idx = state.findIndex((a) => { return a.id === action.payload })
    //console.log(num);
    ++state[idx].amount;
  },
  export let addCount, sortName, addItem, subtractCount, removeItem} = cart.actions
<button onClick={() => dispatch(subtractCount(v.id))}>-</button>
{v.amount}
<button onClick={() => dispatch(addCount(v.id))}>+</button>
```

Store.js export

상품 수량 변경 총 주문 금액 계산 기능을 구현하였습니다.

Add Count와 subtract Count 으로 상품 수량 증감 기능 을 추가했습니다.

003 Website Building

Code Implement6-1



The screenshot shows a shopping cart page for the Beatitude website. The cart contains three items: a rice key ring (¥23,000), a rice ball (¥18,000), and a rice ball rice key ring (¥21,800). The total amount is displayed as 75,800 KRW. The page includes navigation links for HOME, DETAIL, LOGIN, and CART, as well as links for Terms and Conditions, Privacy Policy, and Customer Service. There are also links for About Us, Magazine, Instagram, and SUPPORT.

Use Effect와 calculate Total Sum 함수를 사용하여 장바구니 해당 상품의 총주문 금액을 계산하였고 배송비는 상품이 있을 때만 추가 되도록 하였습니다.

Format Price를 사용하여 화폐단위와 콤마를 추가 하였습니다.

결제금액에는 상품 금액과 배송비의 합계가 보이게 됩니다.

```
const [shopping, setShopping] = useState(3000);
const [totalOrderAmount, setTotalOrderAmount] = useState(0); // 총 주문금액 상태 추가
const calculateTotalSum = () => {
  let sum = 0;
  for (let v of cart) {
    sum += v.price * v.amount;
  }
  return sum;
}
const sum = calculateTotalSum();
setTotalOrderAmount(sum); // 장바구니의 총 주문금액을 업데이트합니다.

if (sum <= 0) {
  setShopping(0);
} else {
  setShopping(3000);
}
}, [cart]); // cart 배열이 변경될 때마다 useEffect가 실행됩니다.

// cart 총 수량
const totalAmount = cart.reduce((acc, item) => acc + item.amount, 0);
<p>총 주문금액 : {formatPrice(totalOrderAmount)}</p>
<p>배송비 : {formatPrice(shopping)}</p>
</div>
<h4 className='total_amount'>
  결제금액 : {formatPrice(Number(totalOrderAmount)+shopping)}
</h4>
```

003 Website Building

Code Implement6-2



The screenshot shows the Beatitude website's cart page. The cart contains three items:

- 1x 틀리가듯이 (Ribbon omelet rice key ring) - ¥23,000
- 1x 떡 (Rice cake) - ¥18,000
- 1x 떡하 (Rice ball) - ¥21,800

The total purchase amount is ¥72,800, shipping is ¥3,000, and the total payment amount is ¥75,800. A "CHECKOUT" button is visible.

```
removeItem(state, action) {
  let idx = state.findIndex(a => a.id === action.payload)
  state.splice(idx, 1)
}
export let { addCount, sortName, addItem, subtractCount, removeItem } = cart.actions
<Button Button variant="dark" onClick={() => dispatch(removeItem(v.id))}>X</
  Button>
```

Store.js export

Remove Item 을 사용하여 장바구니 상품삭제 기능 구현했습니다.
Find Index 메서드를 사용하여 해당 ID를 가진 상품을
Splice 메서드를 사용하여 추가된 상품을 배열에서 제거합니다.

003 Website Building

Code Implement7



The screenshot shows the BeAttitude website's login form. At the top, there is a navigation bar with links for HOME, DETAIL, LOGIN, and CART. The main title 'BEATTITUDE' is displayed above the login form. The login form itself has two input fields: '아이디' (ID) containing 'duddk9804' and '비밀번호' (Password) containing 'password'. Below the inputs is a 'Log In' button. At the bottom of the page, there are links for Terms and Conditions, About Us, Contact, Magazine, Instagram, Shop Instagram, and SUPPORT. The SUPPORT section includes a message about collaboration and contact email addresses.

로그인 폼 입력에 대한 유효성 검사하는 코드를 구현했습니다.

chkPW함수는 이메일 주소와 비밀번호 유효성을 검사합니다.

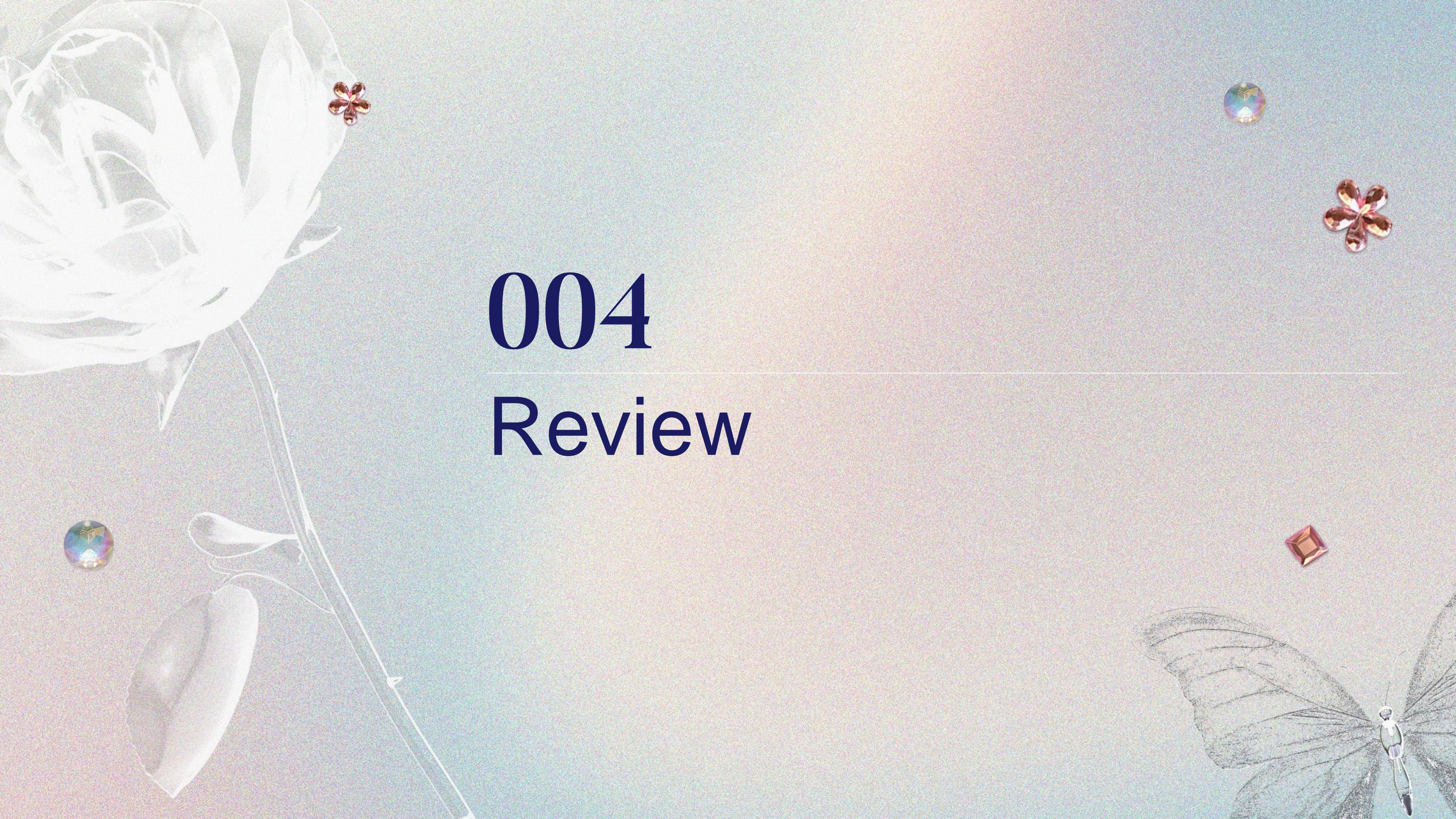
유효성 검사에 적합하면 로그인 완료, 반대의 경우 경고 메시지를 표시합니다.

```
function chkPW() {
    const idIv = id.current.value;
    var pw = pwref.current.value;
    var num = pw.search(/\d/g);
    var eng = pw.search(/[a-z]/ig);
    var spe = pw.search(/[^~!@#$%^&*|\u2000'\u2000';:\u2000/?]/gi);

    if(!emailRegEx.test(idIv)){
        alert('이메일 형식에 맞게 다시 입력해 주세요');
        id.current.focus();
        return false
    }

    if (pw.length < 8 || pw.length > 20) {
        alert("비밀번호는 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 || spe < 0) {
        alert("비밀번호는 영문,숫자, 특수문자를 혼합하여 입력해주세요.");
        return false;
    } else {
        alert("로그인 완료");
        return true;
    }
}

<input type='text' name='userid' placeholder='Id' ref={id}></input><br/>
<div className='id-password'>비밀번호</div>
<input type='password' name='userpassword' ref={pwref} placeholder='password' id='password'></input><br/>
<Button variant="dark" size="lg" onClick={chkPW}>로그인</Button>
```



004

Review



Retrospective

웹페이지를 제작할 때마다 html, CSS, Javascript만 사용했었는데 react를 사용해 보니

편리하고 장점이 많은 것 같습니다. 우선 Bootstrap의 라이브러리를 가져오면

디자인 요소와 반응형도 직접 만들지 않고 복사만으로 사용할 수 있어서 작업할 때

편리했습니다. 제작해야 할 웹 페이지 수가 많을 때나 데이터 변경이 많은

작업을 할 때 사용하면 작업의 능률이 오를 것 같습니다. 구직사이트를 확인해 보면

조건이나 우대사항에 react 사용 프론트엔드 공고를 여러 번 봤는데

저도 그 조건에 맞게 react를 더 공부 하고 싶습니다.



Thank
You