

# PROJECT PROPOSAL

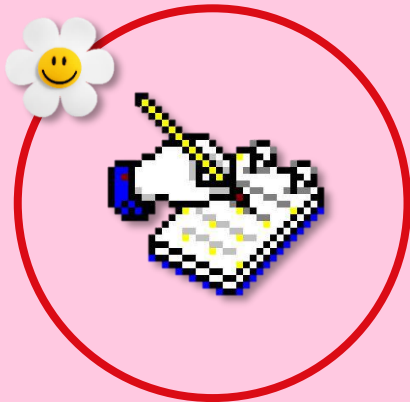


# WIGGLE WIGGLE WEBSITE

[ By. Lim Ye Rang ]



# CONTENT



**01**

**BRAINSTORMING  
BRAND STORY  
MARKET RESEARCH**



**02**

**PERSONAS  
WIREFRAME  
PROTOTYPE**



**03**

**UI DESIGN GUIDE  
CODE  
IMPLEMENTATION**



**04**

**EVALUATION**

# PROJECT SCHEDULE



## APRIL

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

### 1 WEEK



- 주제선정 /자료조사
- 메인 와이어프레임
- 메인페이지 코딩

### 2 WEEK



- 메인페이지 코딩

### 3 WEEK



- 서브 와이어프레임
- 서브페이지 코딩
- 포트폴리오



# BRAINSTORMING



레트로

문구

문구

레트로

편리

귀여움

키치

장소

일상

실용적



# BRAND STORY

wiggle wiggle®

- 🌻 2014년 디자인회사 아트웨어에서 출시.
- 🌻 화려한 비비드컬러의 색감, 유니크한 디자인, 레트로 스타일로 전개.
- 🌻 올리브영, 카카오프렌즈, 앱코 등 대기업, 여러 브랜드와의 협업과 서울 디자인 페스티벌 & 아트 행사 등 참여로 디자인 영역을 확대.



# MARKET RESEARCH

포토뉴스

신세계 센텀시티에 '위글위글' 팝업...MZ세대 '바글바글'

아이의 포토존으로... 센텀에 들어선 '위글위글 피크닉'

위글위글, 카카오프렌즈와 협업 굿즈 출시

이디야커피X위글위글, 콜라보 굿즈 출시

“ MZ세대를 겨냥한 트렌드  
한 아이템”

“ 의류, 주방용품, 아웃도어  
등 제품의 다양성”

“ 다양한 브랜드와  
콜라보로 차별성”

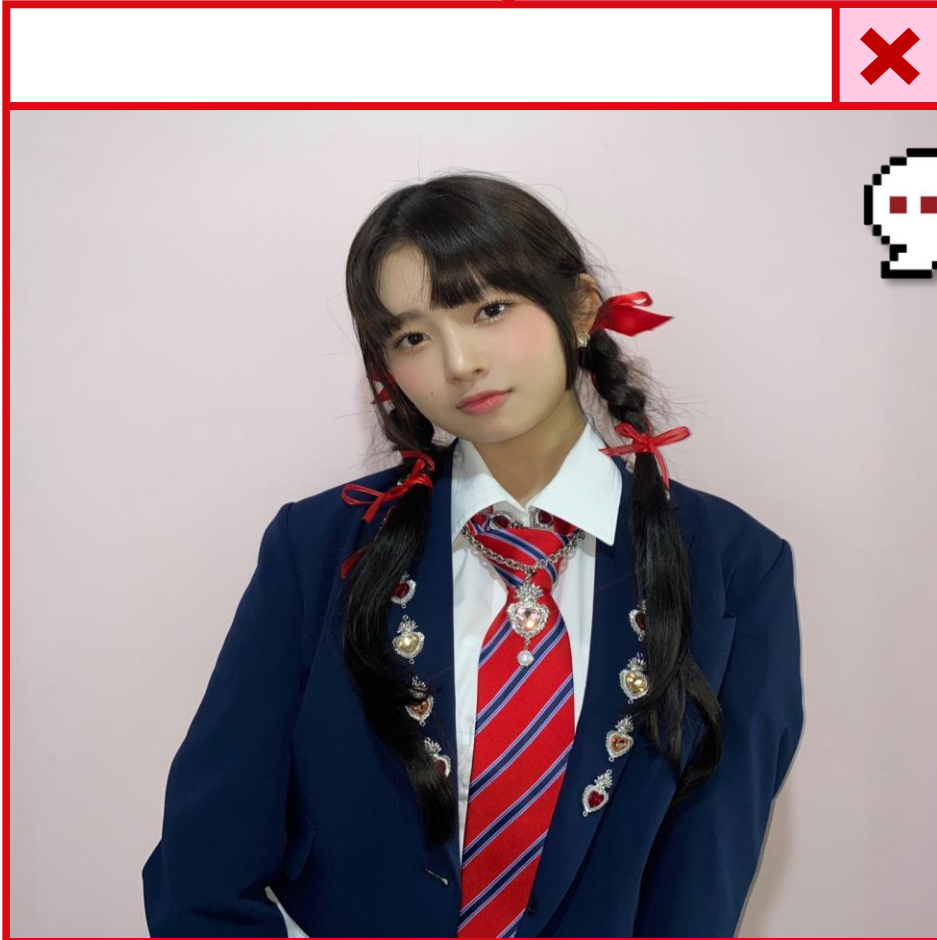
“ 오프라인 스토어로  
시각적 효과 극대화”



OFFLINE STORE



# PERSONAS



이름 : 이민영

나이 : 19

직업 : 방송연예과 3학년 고등학생

취미 : 영화보기, 액세서리 만들기, 영상편집하기

라이프 스타일 : 예술고등학교에 재학 중이며 학교 생활과 병행하며 가끔씩 소소하게 액세서리를 직접 만들어 블로그 마켓에 판매 중이다. 이런 자신의 일상을 기록하는 유튜브를 운영 중이며 키치한 것을 좋아해 비슷한 아이템을 발견 시 소개하는 영상을 종종 올리고는 한다. 학교생활로 바빠 영상 올린지가 꽤 뜸해져 키치 아이템 소개 영상을 찍고자 한다.

“위글위글에서 여름맞이 키치한 비치웨어가 나왔다고 들었어. 어떤 아이템인지 직접 보고 싶어. 오프라인 매장 위치와 방문 전 상품을 미리 온라인으로 구경 후 영상에 기록하고 싶어”



# PERSONAS



이름 : 한유인

나이 : 25

직업 : 유치원 교사

취미 : 뜨개질, 그림그리기, 독서, 다이어리 쓰기

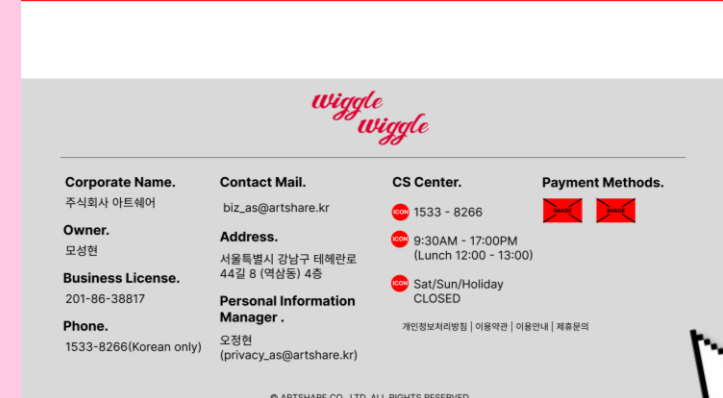
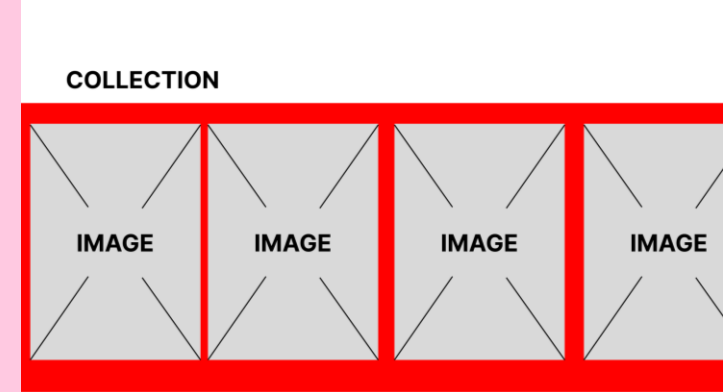
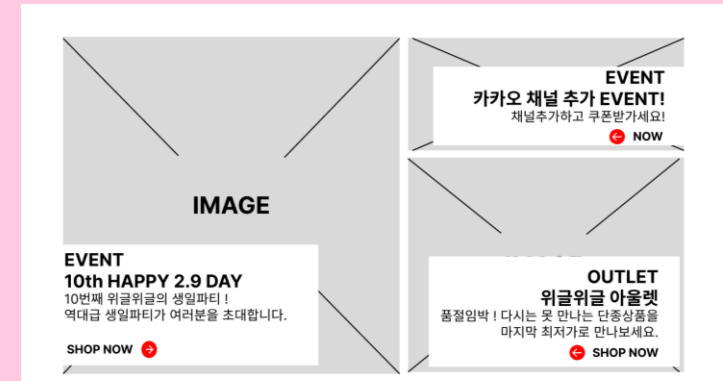
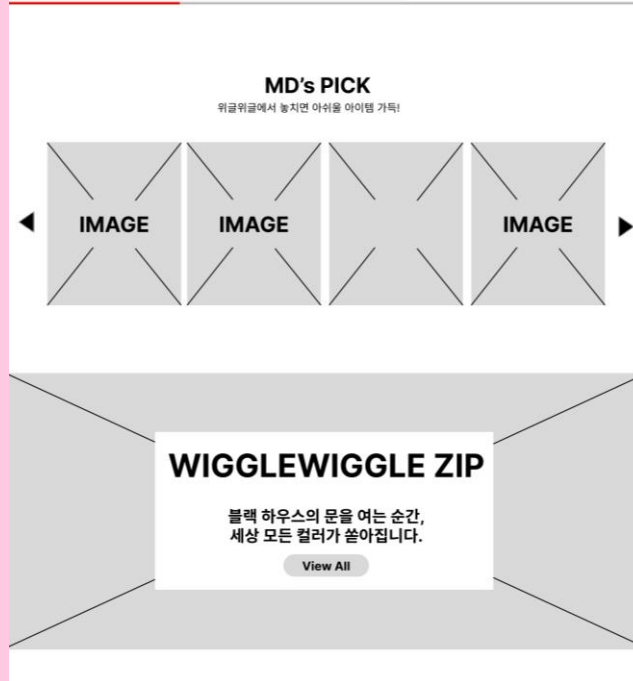
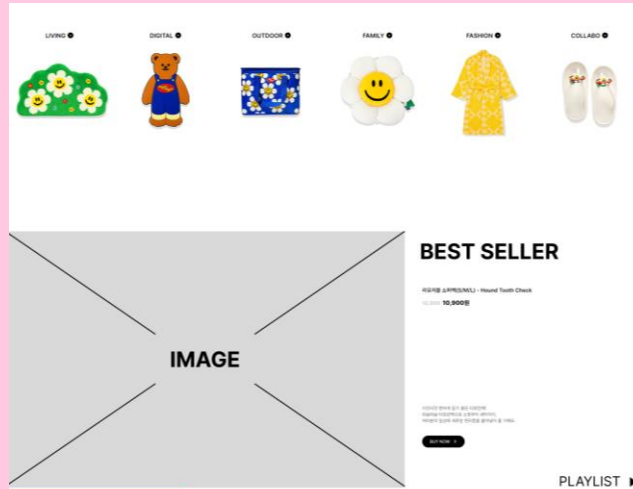
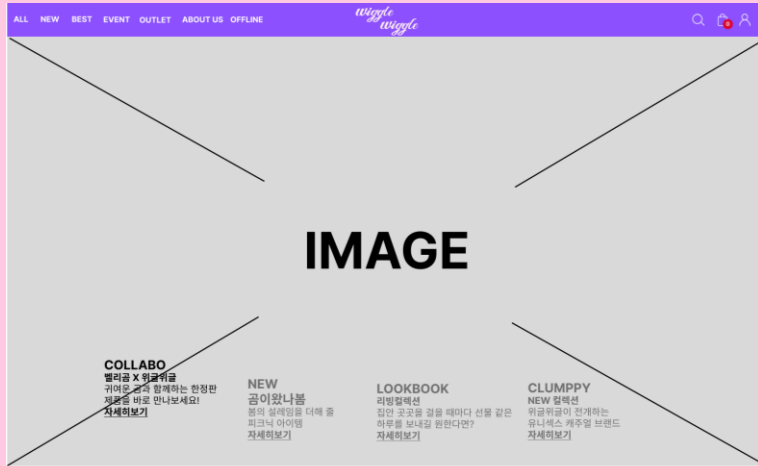
라이프 스타일 : 5-7세를 돌보는 유치원의 햇살반 교사로 임명되어 아이들을 맞이하기위한 준비로 바쁜 하루를 보내고 있다. 꼼꼼하고 계획하는 것을 좋아해 작은 것 하나하나 신경쓰는 타입이다. 아이들 수첩을 작성하거나 서류 작성에 필요한 펜, 노트 등 문구 아이템을 찾는 중이다.

“나는 귀여운 아이템을 수집하는 것을 좋아해. 디자인에도 신경을 쓰고 실용적인 사용할 수 있는 문구 아이템이 어디 없을까?”





# WIREFRAME(MAIN)



원페이지 고정

ALLNEWBESTEVENTOUTLETABOUT USOFFLINE

wiggle wiggle

Search, Cart, User icons

IconDetailIconReviewIconQ&AIconRecommendation

All > 문구 > 펜/노트

하드 커버 스프링 노트 -Spring Garden

16,500원14,800원10%

옵션[필수] 옵션을 선택해 주세요

배송비50,000원 이상 구매 시 무료배송  
기본 3,000원

총 상품 금액14,800원구매하기

오른쪽 상세페이지만  
스크롤로 이동

IconDetailIconReviewIconQ&AIconRecommendation

All > 문구 > 펜/노트

하드 커버 스프링 노트 -Spring Garden

16,500원14,800원10%

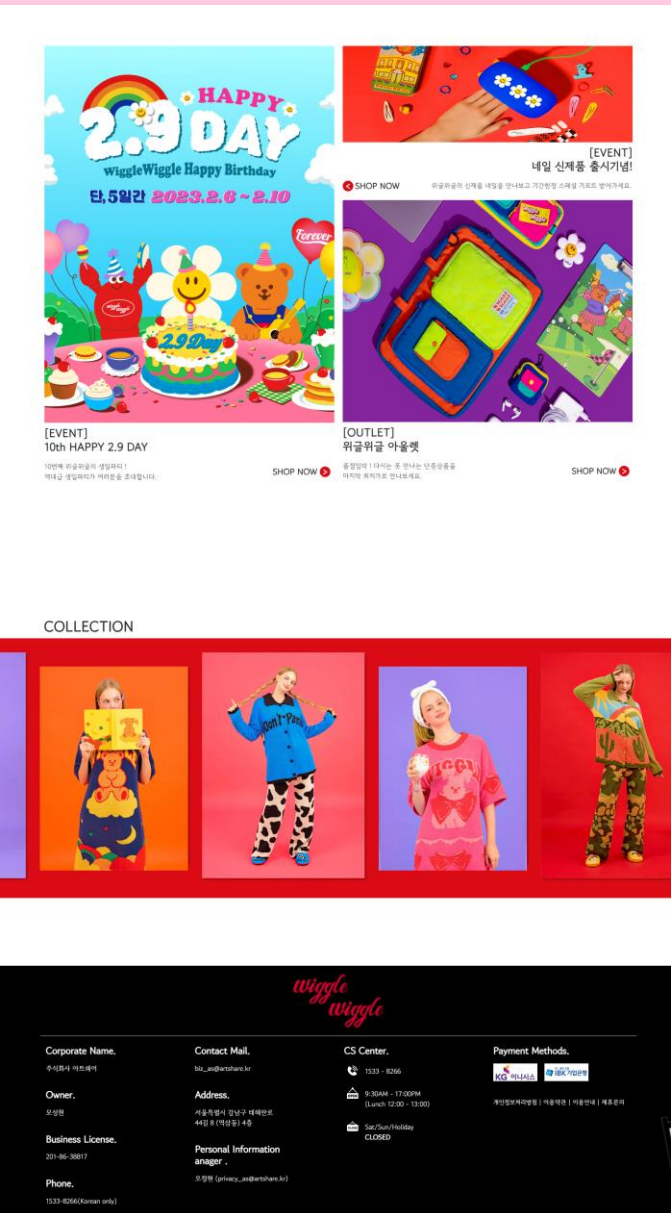
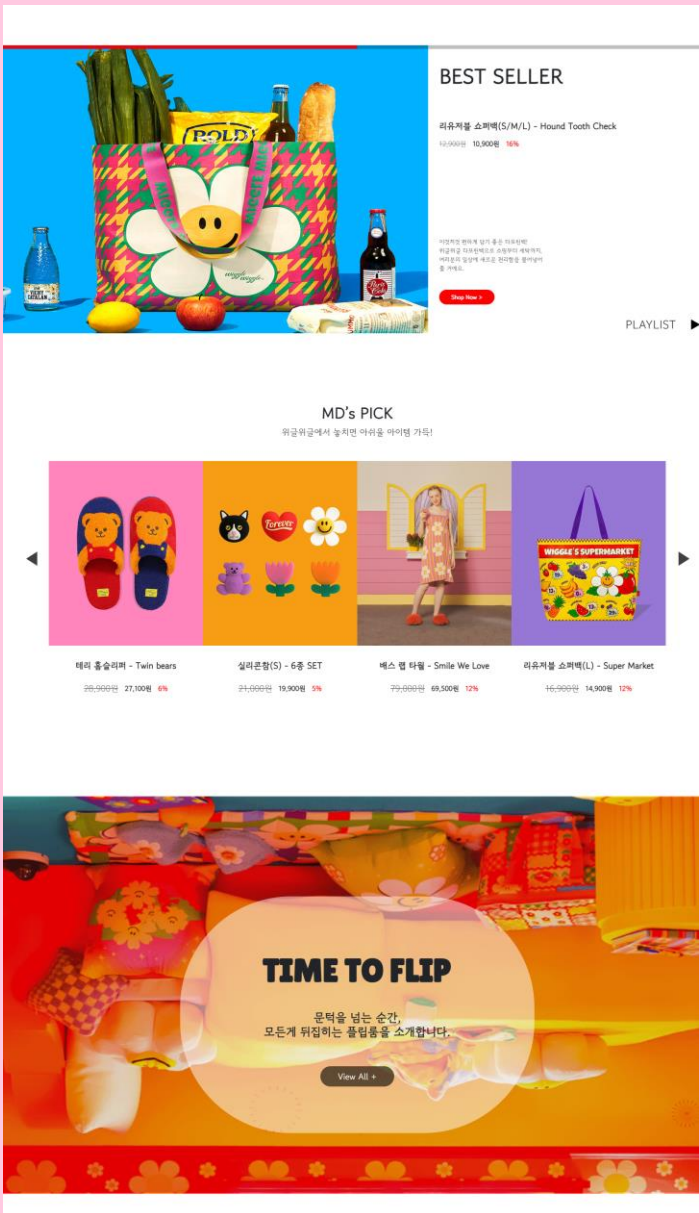
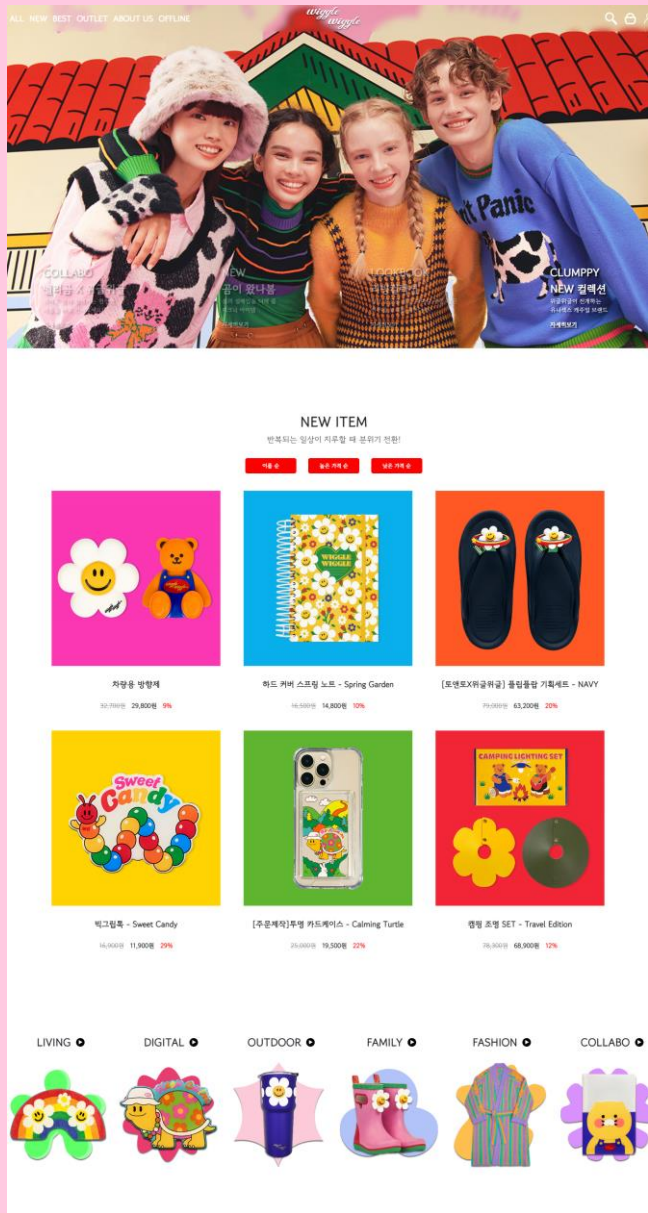
옵션[필수] 옵션을 선택해 주세요

배송비50,000원 이상 구매 시 무료배송  
기본 3,000원

총 상품 금액14,800원구매하기

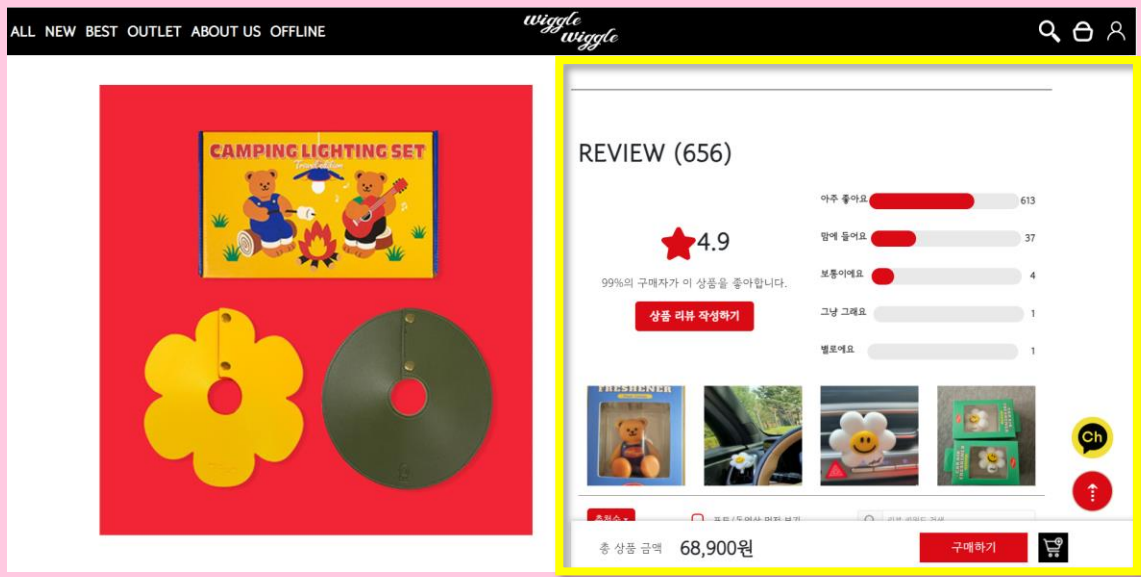
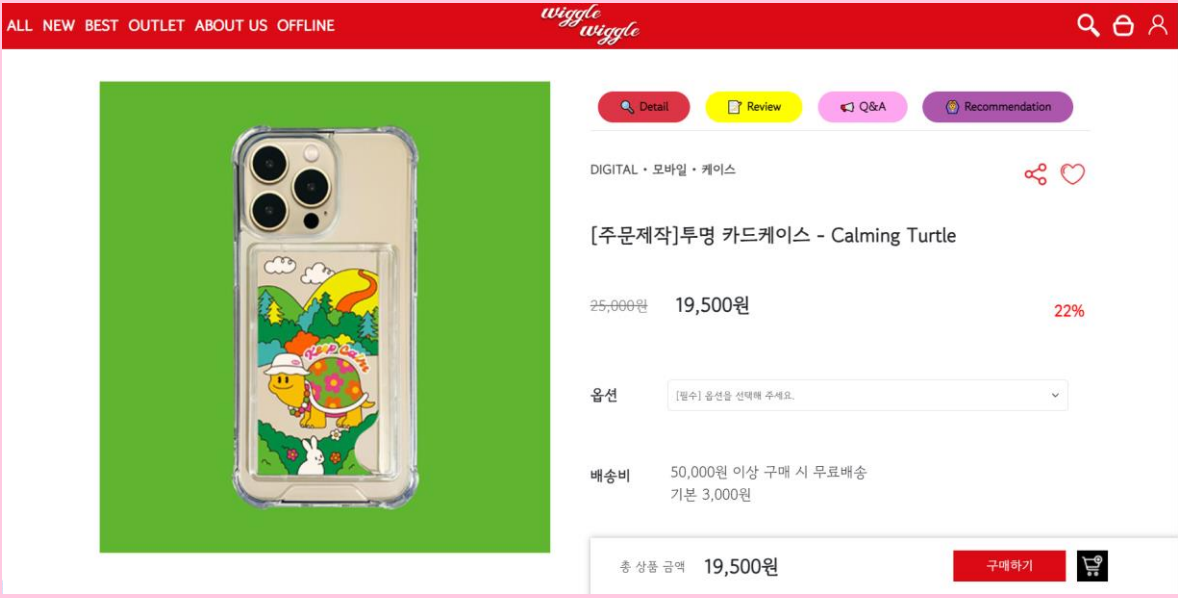


# PROTOTYPE(MAIN)

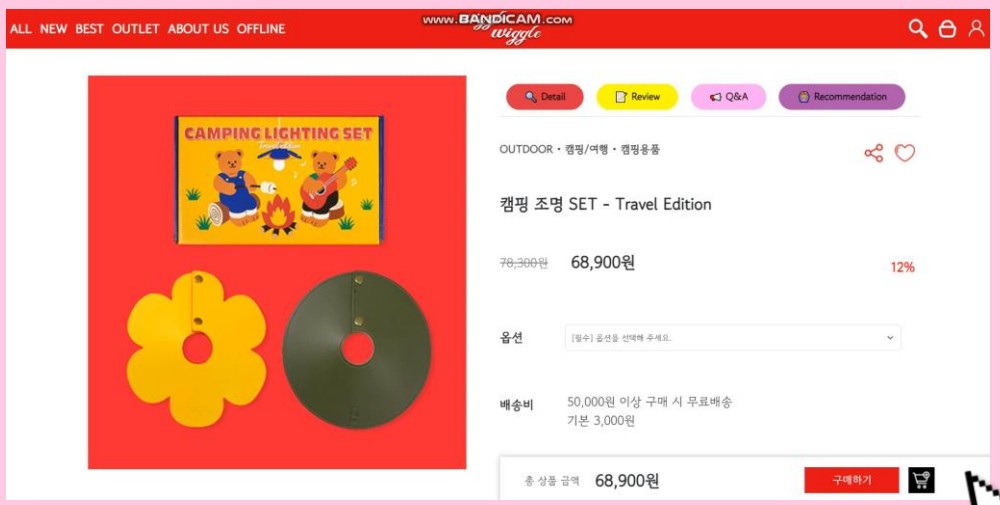




# PROTOTYPE(SUB)



오른쪽 상세페이지 스크롤



PLAY



## COLOR



- 🌸 BackgroundColor : #ffffff
- 🌸 SubColor : #db0b15
- 🌸 FontColor : #0000

## TYPOGRAPHY



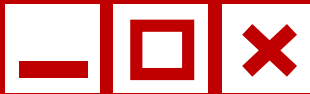
- 🌸 Korea Fonts: Lilita One
- 🌸 English Fonts: Gowun Dodum

## COLOR

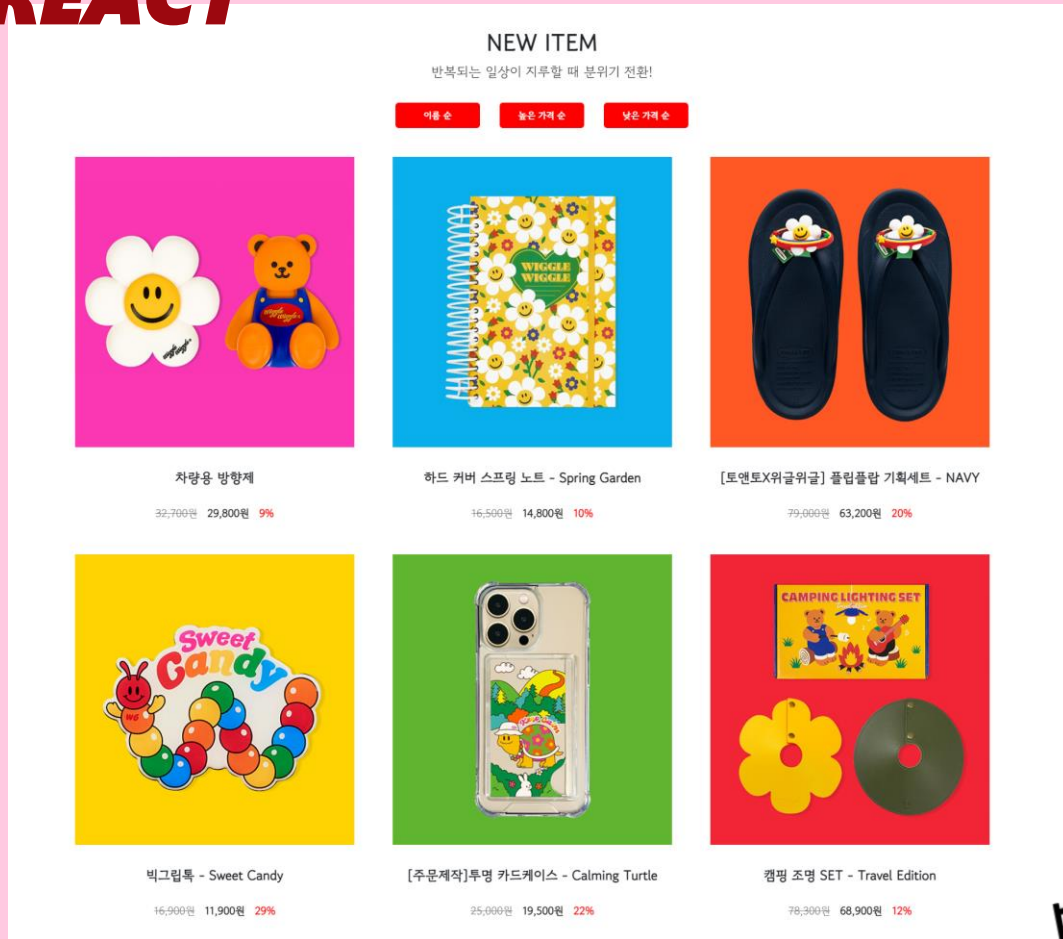




# CODE IMPLEMENTATION(MAIN)



## REACT



Product 페이지를 만들어 데이터 정보를 전달받아 맵으로 돌려 저장된 데이터를 전송받아 구현하였습니다.



이름순, 높은 가격순, 낮은 가격순 버튼 클릭시 각각 정렬이 되도록 카피하여 sort() 함수를 이용하여 구현하였습니다.

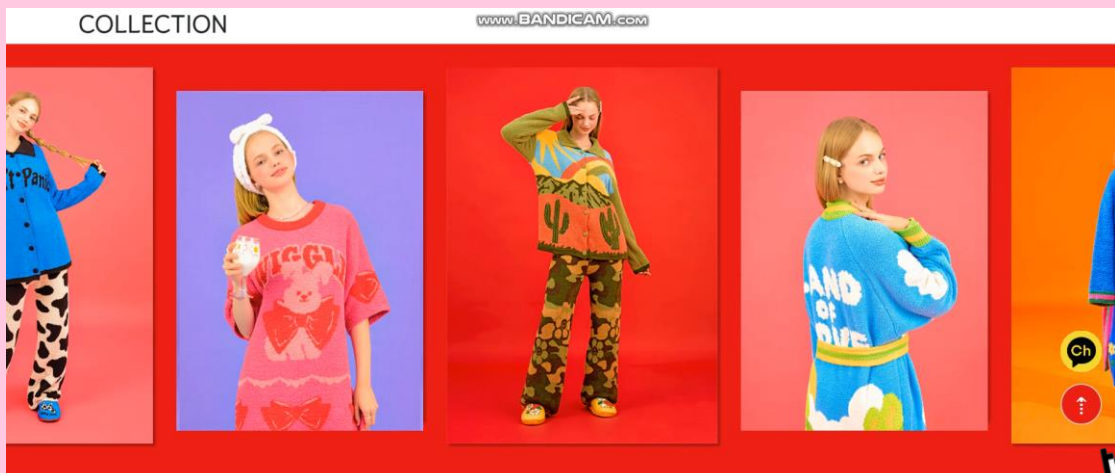
```
<Button variant="danger" onClick={() => {  
  let copy = [...new_item].sort((a, b) => (a.sale < b.sale) ? 1 : -1);  
  setNew_item(copy);  
  let no_copy = [];  
  for (let i in copy) {  
    no_copy.push(copy[i].id);  
  }  
  setNo(no_copy);  
}}>높은 가격 순</Button>{' '}  
<Button variant="danger" onClick={() => {  
  let copy = [...new_item].sort((a, b) => (a.sale > b.sale) ? 1 : -1);  
  setNew_item(copy);  
  let no_copy = [];  
  for (let i in copy) {  
    no_copy.push(copy[i].id);  
  }  
  setNo(no_copy);  
}}>낮은 가격 순</Button>{' '}
```

```
<Container className='New'>  
  <Row>  
    {  
      new_item.map((new_item, i) =>  
        <Product new_item={new_item} key={i} i={i} no={no} />  
      )  
    }  
  </Row>
```

```
const Product = (props) => {  
  let navigate = useNavigate();  
  let {new_item, i, no} = props;  
  const {title, imgUrl, price, sale, discount} = props.new_item;  
  return (  
    <Col sm={6} xl={4}>  
      <Nav.Link onClick={()=> navigate('/detail/' + no[i])}>  
        <div>  
          <img src={process.env.PUBLIC_URL + imgUrl} width="100%" />  
          <div className='textbox'>  
            <h4>{title}</h4>  
            <div className='sub_textbox'>  
              <p>{price}</p><div>  
                <b>{sale.toString().replace(/\B(?!(\d{3})+(?!))/g, ",")}원</b><div>  
                  <span>{discount}</span></div>  
            </div>  
          </div>  
        </Nav.Link>  
      </Col>
```



# CODE IMPLEMENTATION(MAIN)



**PLAY**



갤러리 형식의 롤링배너를 구현하기 위해 배열에 src로 이미지를 저장하고 슬라이드박스를 .concat() 함수를 이용하여 복제해 이어붙였으며 onMouseEnter로 마우스 올릴 시 멈추는 배너를 구현하였습니다.

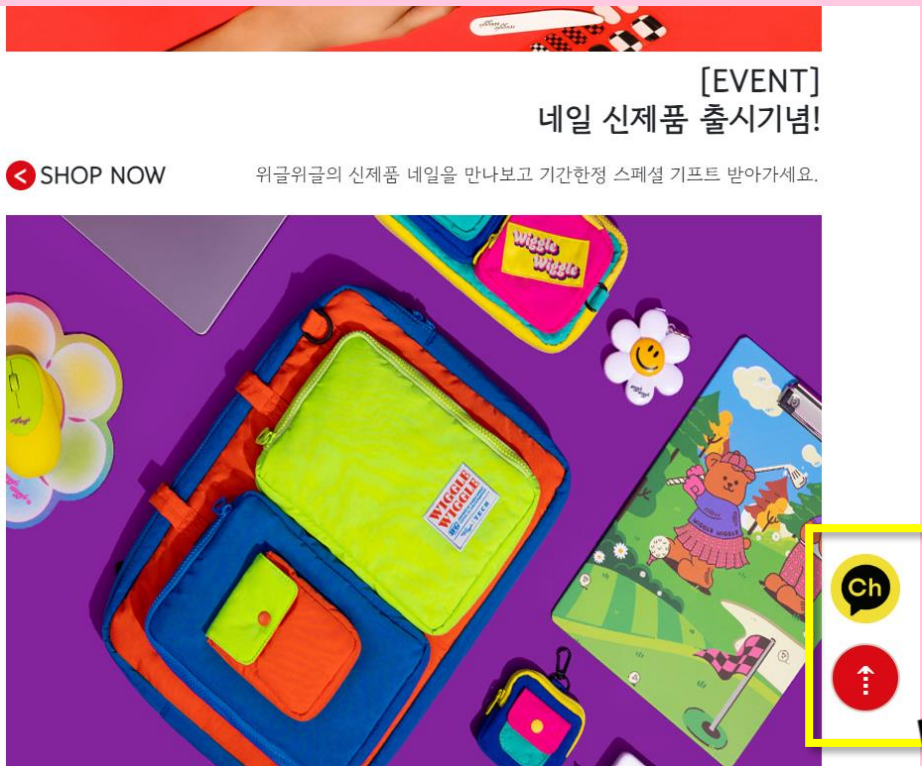
```
const slide = [
  { src: "/img/collection1.jpg" },
  { src: "/img/collection3.jpg" },
  { src: "/img/collection4.jpg" },
  { src: "/img/collection2.jpg" },
  { src: "/img/collection5.jpg" },
  { src: "/img/collection7.jpg" },
  { src: "/img/collection6.jpg" },
  { src: "/img/collection9.jpg" },
  { src: "/img/collection8.jpg" },
  { src: "/img/collection11.jpg" },
  { src: "/img/collection10.jpg" },
];
```

```
/*무한 자동 슬라이드*/
const onRun = () => setAnimate(true);
/*마우스 커서 시 stop*/
const onStop = () => setAnimate(false);

return (
  <div className="Rolling">
    <h2>COLLECTION</h2>
    <div className="slide_container">
      <ul
        className="slide_rolling"
        onMouseEnter={onStop}
        onMouseLeave={onRun}
      >
        <div
```

```
<div
  className={"slide original".concat(
    animate ? "" : " stop"
  )}
>
  {slide.map((g, i) => (
    <li
      key={i}
      className={i % 2 === 0 ? "img_box1" : "img_box2"}
    >
      <div
        className="slide_box"
        // style={{ backgroundImage: g.src }}
      >
        <img src={process.env.PUBLIC_URL + g.src} alt="" width="100%" />
      </div>
    </li>
  ))}
</div>
<div
  className={"slide clone".concat(animate ? "" : " stop")}
>
  {slide.map((g, i) => (
    <li
      key={i}
      className={i % 2 === 0 ? "img_box1" : "img_box2"}
    >
      <div
        className="slide_box"
        // style={{ background: g.src }}
      >
        <img src={process.env.PUBLIC_URL + g.src} alt="" width="100%" />
      </div>
    </li>
  ))}
</div>
```

# CODE IMPLEMENTATION(MAIN)

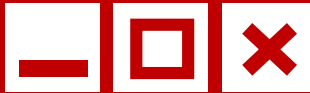


스크롤 시 탑 버튼이 나타나고 내릴때마다 버튼 클릭시 제일 맨 위로 올라오는 화면을 구성하기 위해 useEffect, useState 이용하였고 scroll 값을 500을 지정해 스크롤을 500이상 내려올 때 버튼이 등장하고 논리연산자 && 로 나타나고 사라지는 화면을 구현하였습니다.

```
export default function ScrollToTop() {  
  
  const { pathname } = useLocation();  
  const [showButton, setShowButton] = useState(false);  
  
  const scrollToTop = () => {  
    window.scrollTo({  
      top: 0,  
      behavior: 'smooth'  
    })  
  }  
  
  useEffect(() => {  
    const handleShowButton = () => {  
      if (window.scrollY > 500) {  
        setShowButton(true)  
      } else {  
        setShowButton(false)  
      }  
    }  
  
    console.log(window.scrollY)  
    window.addEventListener("scroll", handleShowButton)  
    return () => {  
      window.removeEventListener("scroll", handleShowButton)  
    }  
  }, [])  
  
  useEffect(() => {  
    window.scrollTo(0, 0);  
  }, [pathname]);  
  // return null;  
  return showButton && (  
    <div className="scroll_Cover">  
        
      <button id="top" onClick={scrollToTop} type="button" >↑</button>  
    </div>  
  )  
}
```



# CODE IMPLEMENTATION(SUB)



## CART

상품명	할인율	할인가	수량	합계	선택
차량용 방향제	9%	29,800원	- 1 +	29,800원	삭제하기
하드 커버 스프링 노트 - Spring Garden	10%	14,800원	- 2 +	29,600원	삭제하기

1개 이하



클릭 시 →



localhost:3000 내용:  
1개 이상부터 구매가능합니다.

확인



Redux를 사용해 함수와 변수를 전달받아 각각 함수를 지정해 수량값이 하나씩 증가, 감소가 가능하도록 구현하였습니다.

29,800원

삭제하기

클릭 시

29,800원

삭제하기

29,600원

삭제하기



removeItem이라는 함수이름을 지정해 “삭제하기” 클릭 시 각 카트에 담은 상품을 삭제해주도록 구현하였습니다.

```
reducers: {
  addCount(state, action) {
    let idx = state.findIndex((a) => {
      return a.id ===
        action.payload
    })
    ++state[idx].amount;
  },
  removeCount(state, action) {
    let idx = state.findIndex(a => a.id === action.payload)
    state[idx].amount--
    if(state[idx].amount < 1){
      alert('1개 이상부터 구매가능합니다.');
```

```
      state[idx].amount = 1;
    }
  },
  sortName(state) {
    state.sort((a, b) =>
      | a.item > b.item ? 1 : -1,);
  },
  addItem(state, action) {
    if (action.payload.type === '0_add') {
      let idx = state.findIndex(a => a.id === action.payload.id)
      if (idx >= 0) {
        state[idx].amount++;
      } else {
        state.push(action.payload);
      }
    } else {
      return state.push(action.payload)
    }
  },
  removeItem(state, action) {
    let idx = state.findIndex(a => a.id === action.payload)
    alert('선택한 상품을 삭제합니다.');
```

```
    state.splice(idx,1)
  }
}
export let { removeCount, addCount, sortName, addItem, removeItem } = cart.actions

export default configureStore({
  reducer: {
    | cart: cart.reducer
  }
})
})
```



# CODE IMPLEMENTATION(SUB)



상품명	할인율	할인가	수량	합계
차량용 방향제	9%	29,800원	<div><div>-</div><div>1</div><div>+</div></div>	29,800원

↓ 수량 클릭 시 값을 전송

총 상품금액	총 배송비	결제예정금액
29,800원	3,000원	32,800원

↓ 50,000원 이상 시 배송비 삭제

총 상품금액	총 배송비	결제예정금액
59,400원	0원	59,400원
5만원 이상 구매 시 무료배송		

```
// 총합계
function Total_price(){
  let idx = 0;
  for(let v in state.cart){
    |   idx += state.cart[v].sale * state.cart[v].amount
  }
  return idx
}
let [Shbag,setShbag] = useState(3000);

useEffect(()=>{
  function Shbag(){
    |   if(Total_price() >= 50000){
    |     setShbag(0);
    |   }else{
    |     setShbag(3000);
    |   }
  }
  Shbag();
});
```



기본 배송비 3,000원을 50,000원 이상일 때 무료배송으로 자동으로 입력받기 위해 state 값을 지정해줬으며 각 상품을 추가 시 자동으로 합계를 계산해주도록 구현하였습니다.





## LOGIN

카카오 간편 회원가입하면  
3,000원 할인 쿠폰 즉시 지급!

\*플친 쿠폰은 ID당 1회 지급

카카오계정 로그인

TALK

3,000원  
플친 쿠폰

10%  
COUPON

신규가입시  
10% 쿠폰 증정

FREE

5만원 이상  
무료배송

아이디

ID

비밀번호

PASSWORD

로그인

N 네이버 로그인

잘못된 값 입력 시  
경고창

localhost:3000 내용:  
아이디(이메일)을 다시 입력해주세요.

정상적인 값 입력 시  
경고창

localhost:3000 내용:  
로그인되었습니다.

```
//id, password
function chkPW() {
  const idIv = id.current.value;
  const passIv = Passw.current.value;
  const reg = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{8,}$/;
  const EmailReg = /^[([w-]+(?:\.[w-]+)*)@((?:[w-]+\.)*[w]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/;

  if (!EmailReg.test(idIv)) {
    alert('아이디(이메일)을 다시 입력해주세요.');
```



정규표현식을 사용하여 로그인 페이지에 아이디, 비밀번호 입력 시 value() 속성을 받아와 8자 이상, 숫자/대문자/소문자/특수문자를 포함하여 입력이 가능한 로그인 Form을 구현하였습니다.



## 프로젝트를 마치면서



마지막 프로젝트를 진행하면서 React를 처음 접해봤습니다. 지난 프로젝트에서 사용했던 방식과는 약간 달라 처음 프로젝트를 생성하는 과정에서 생각한대로 되지 않아 어려움이 있었습니다. 하나씩 진행하다보니 이런 원리로 이뤄진다는 것을 알게 되면서 구상한대로 구현될 때 뿌듯함을 느낄 수 있었습니다. 프로젝트를 완성했어도 리액트를 완전히 이해했다고 보기에는 힘들지만 조금 더 공부를 하고 이해하면 지금보다 더욱 편리하게 사용할 수 있을 것 같습니다. 마지막 작업물이다보니 더 욕심내게 되었는데 지난 프로젝트부터 작업 속도가 향상되지않아 아쉬움이 계속 남았습니다. 개선하기위해 개인적인 공부가 좀 더 필요하고 노력이 필요하다고 느꼈습니다.

