

# 포트폴리오 소개

## DOREMI

기업 홈페이지에서 일반적으로 사용하고 있는  
조직도, 게시판, 미디어(동영상/갤러리), 지도, 회원가입 메뉴로 구성하여 제작

### 사용 기술



REACT



CSS3

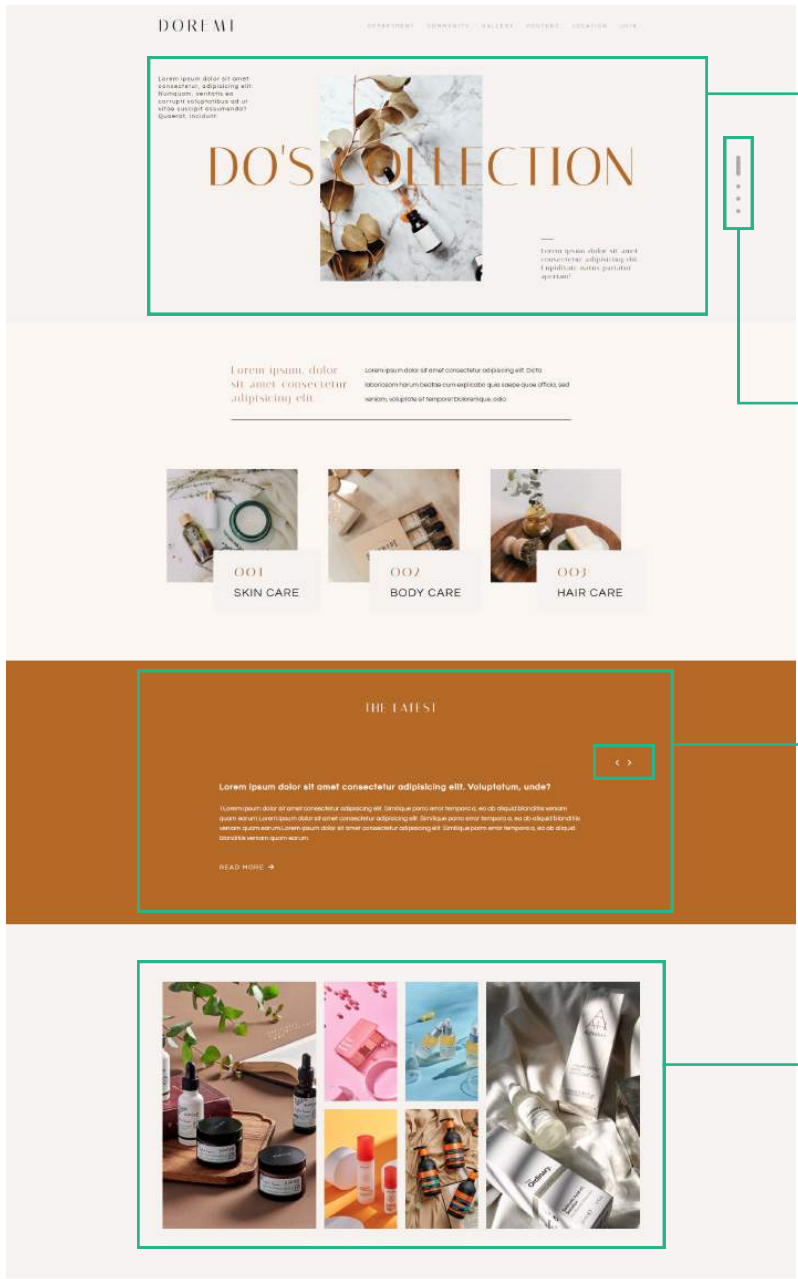


SASS(SCSS)

### 목차

1. 메인 페이지	.....	2 page
2. 반응형 웹 레이아웃	.....	3 page
3. 서브 페이지	.....	4 page
a. Department		
b. Community	.....	5 page
c. Gallery	.....	6 page
i. Flickr API		
d. Youtube	.....	7 page
i. Youtube API		
e. Location	.....	8 page
i. Kakaomap API		
f. Join	.....	9 page

# 메인 페이지



## 1 메인페이지 텍스트 모션

마우스 움직임에 따라  
센터를 기준으로 움직이는 모션 적용

## 2 스크롤 모션

offsetTop으로 각 section마다  
높이값을 구한 다음 스크롤시  
해당 section이 보이면 네비게이션 버튼 활성화

## 3 메인페이지 게시판

- 리덕스를 이용하여 community의 Q&A 내  
용을 동일하게 메인페이지에 표출
- 좌우버튼 클릭시 슬라이드 모션 적용

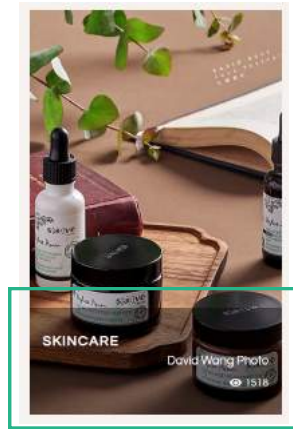
## 4 메인페이지 갤러리

flickr API를 이용하여 갤러리에 저장되어 있는  
이미지를 불러와 표출



## 5 쿠키 팝업

오늘 하루 그만보기 체크시 방문자의 접속시간을  
쿠키 값에 저장하여 24시간 동안 팝업창을 열지않음



마우스오버시  
→ 사진의 제목, 사진소유주 이름,  
조회수 표출

# 반응형 웹 레이아웃

모든 기기에서 동일하게 웹사이트를 이용할 수 있도록  
미디어쿼리를 활용해 브라우저 폭에 따라 각각 desktop, tablet, mobile 버전으로 반응형 처리

미디어 쿼리처리 예시

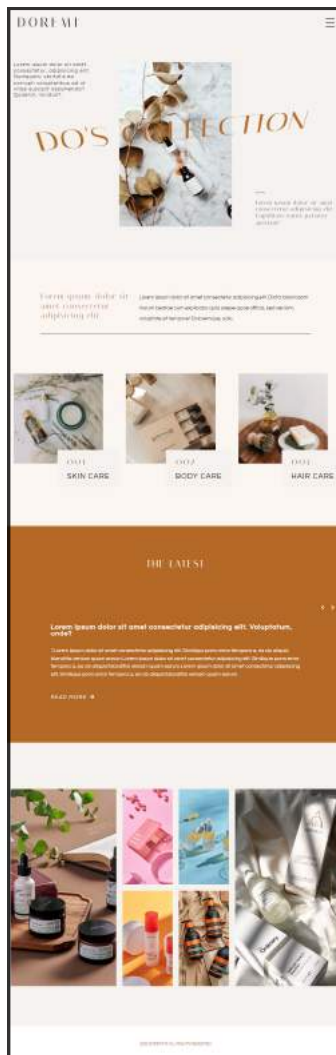
Sass로 변수를 설정한 후 사용

\$laptop: 1025px; → 브라우저 폭이 1025px 미만일 경우 laptop반응형으로 변환

\$tablet: 769px; → 브라우저 폭이 769px 미만일 경우 tablet반응형으로 변환

\$mobile: 461px; → 브라우저 폭이 461px 미만일 경우 mobile반응형으로 변환

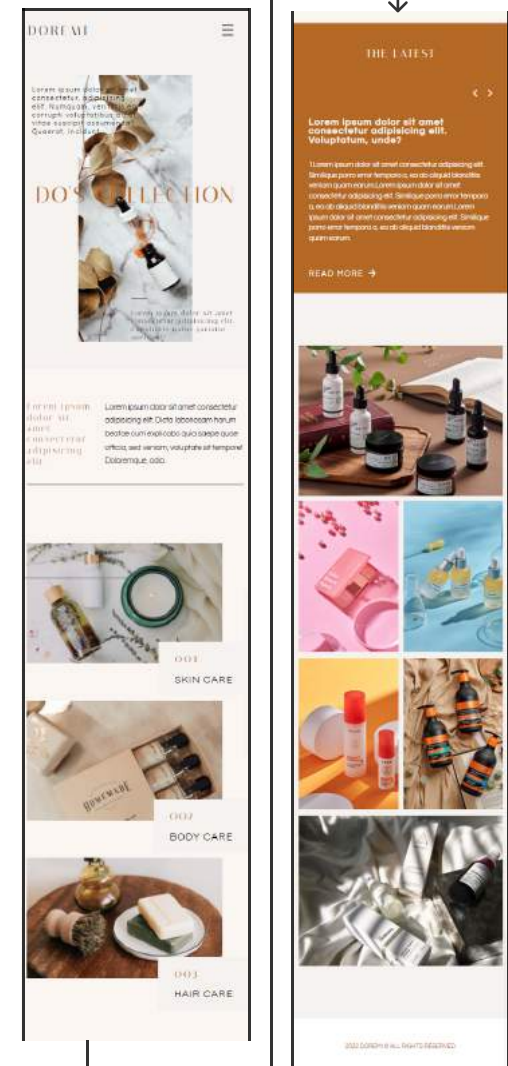
## 1 Desktop



## 2 Tablet



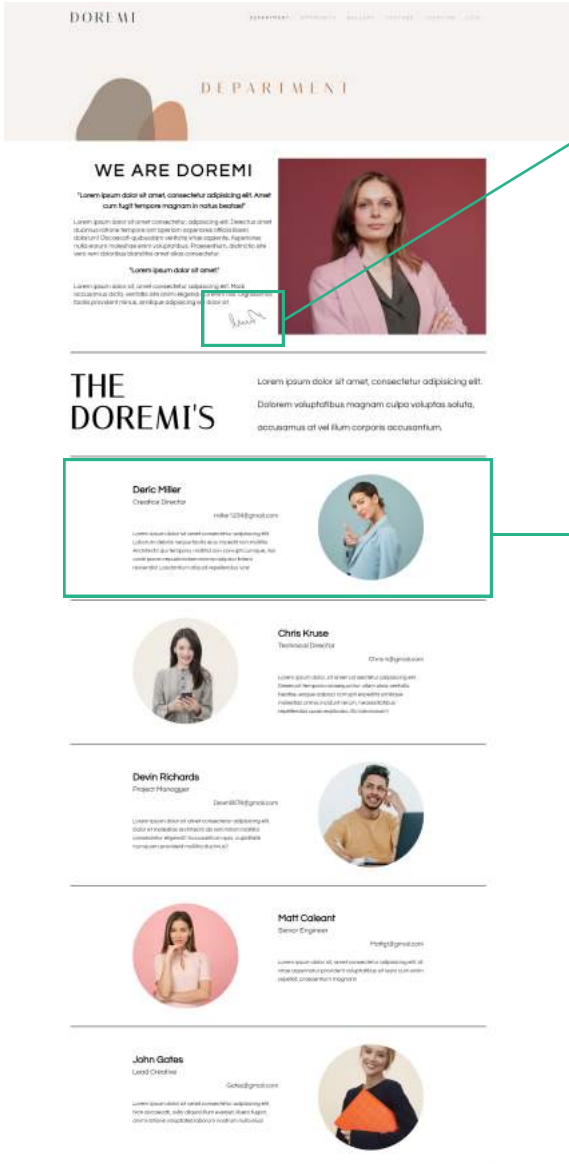
## 3 Mobile



# 서브페이지

# github pages 배포 이슈

## 1 Department 팀원 소개 페이지



### 1. svg 애니메이션

SVG로 path를 만들어 CEO가 결재 사인을 하는 애니메이션을 반복 시킴

### 2. public/dbs/department.json

팀원 소개에 들어갈 팀원 이름, 직급, 이메일, 자기소개, 사진 data값을 json으로 생성

```
"data" : [
  {
    "img": "/img/member1.jpg",
    "name": "Deric Miller",
    "position": "Creative Director",
    "email": "miller1234@gmail.com",
    "introduce": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum debitis neque facilis....",
  },

```

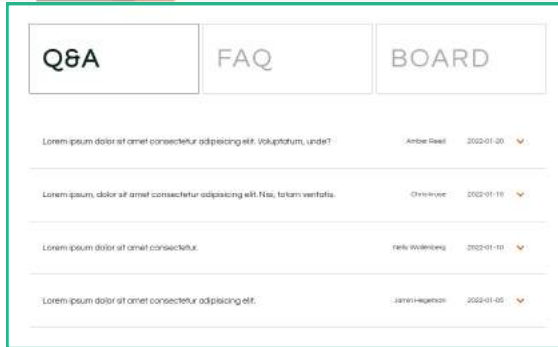
1. 서브 페이지에서 새로고침을 하면 404 에러가 표출됨.
2. github pages에서 SPA를 지원하지 않는 문제로 확인.
3. HashRouter로 변경하니 문제없이 잘 동작함.
4. MIT license의 오픈소스를 참고하여 BrowserRouter를 그대로 사용할 수 있는 방법을 찾아 적용 (참고 : <https://github.com/rafgraph/spa-github-pages>)

- public폴더의 절대경로를 base로 설정
- useState로 상태값을 관리하며 데이터를 계속 호출하지 않도록 useEffect 사용
- axios를 이용하여 데이터를 호출 한 후 map 함수를 사용하여 리스트를 생성

```
const base = process.env.PUBLIC_URL;
const [list, setList] = useState([]);

useEffect(() => {
  axios.get(`${base}/dbs/department.json`).then((json) => {
    setList(json.data.data);
  });
}, []);
```

## 2 Community 게시판 페이지



placeholder text for image



### 1. 게시판 탭 기능

- Q&A, FAQ, BOARD 탭을 누르면 클릭시 각 탭에 해당되는 데이터를 불러와서 표출
- 리스트 클릭시 해당 리스트의 답변이 나타나고, 재클릭시 닫힘

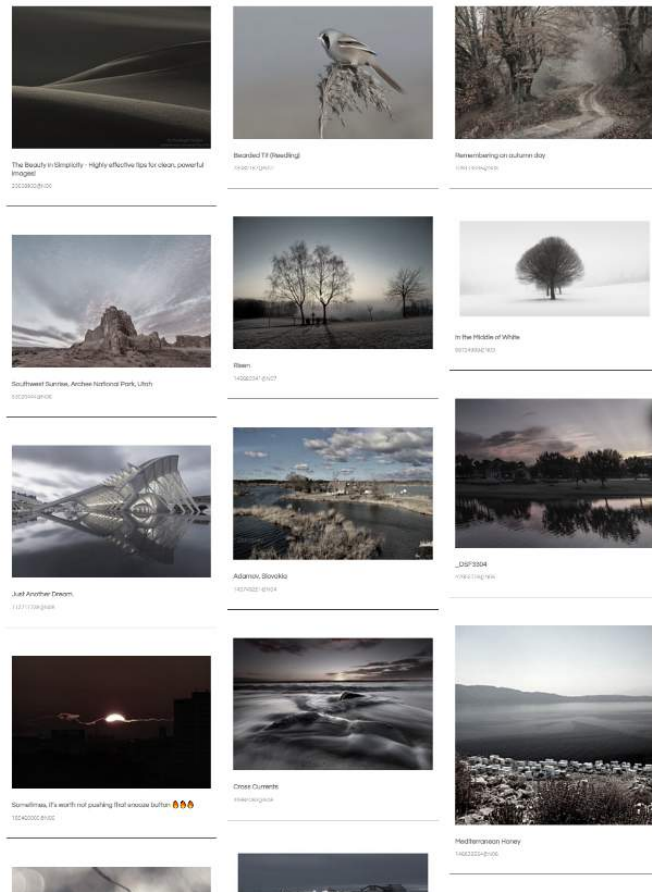
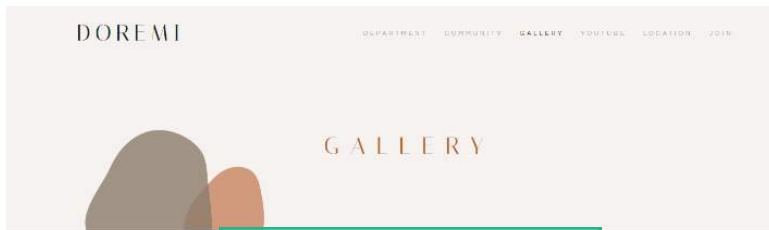


### 2. 메인페이지에 Q&A내용 표출

- react-redux의 useSelector, useDispatch를 이용하여 store에 저장된 Q&A데이터를 가지고와서 표출

```
const community = useSelector((state) => state);
const dispatch = useDispatch();
const fetchCommunity = async () => {
  const response = await axios.get(`${base}/dbs/community.json`).catch((err) => {
    console.error(err);
  });
  dispatch(setCommunity(response.data.data));
};
```





## 1. 검색 기능

- 이미지 로딩이 완료되면 로딩바가 사라지면서 이미지 표출
  - getFlickr(option) 이미지 API를 요청하는 함수
- 옵션의 type을 interest(인기)와 search로 나누어 다른 method로 API를 요청

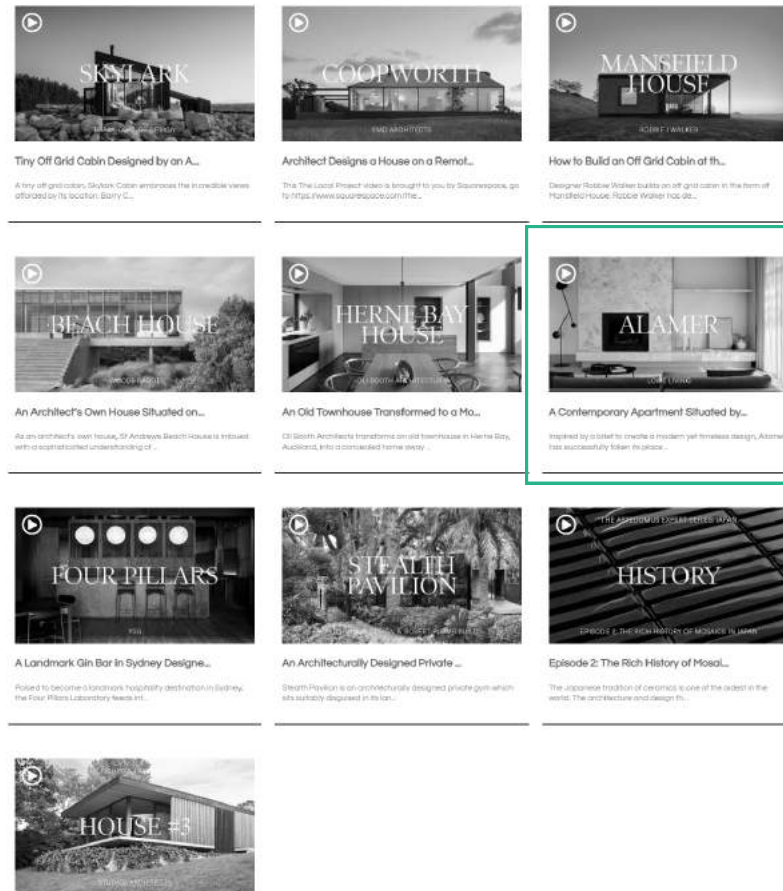
```
getFlickr({ type: "search", tags: tags, count: 50 });
```

```
if (opt.type === "interest") {
  url = `${base}method=${method1}&api_key=${api_key}&per_page=${per_page}&format=json&nojsoncallback=1`;
} else if (opt.type === "search") {
  url = `${base}method=${method2}&api_key=${api_key}&per_page=${per_page}&format=json&nojsoncallback=1&tags=${opt.tags}`;
} else {
  console.error("api요청 타입을 interest, search 중에서 지정하세요.");
}
```

## 2. 팝업 기능

- 이미지 클릭시 썸네일보다 높은 해상도의 이미지가 팝업창으로 생성





## 1. Youtube API

- 개인 플레이리스트에 저장되어있는 동영상 리스트를 요청
- 동영상 제목과 설명글을 원하는 길이만큼 자르고 말줄임(...)을 더하여 표출

```
let tit = item.snippet.title;
let tit_len = tit.length;
let desc = item.snippet.description;
let desc_len = desc.length;
```

```
<h2>{tit_len > 36 ? (tit = tit.substr(0, 36) + "...") : tit}</h2>
<p>{desc_len > 50 ? (desc = desc.substr(0, 100) + "...") : desc}</p>
```



## 2. 팝업 기능

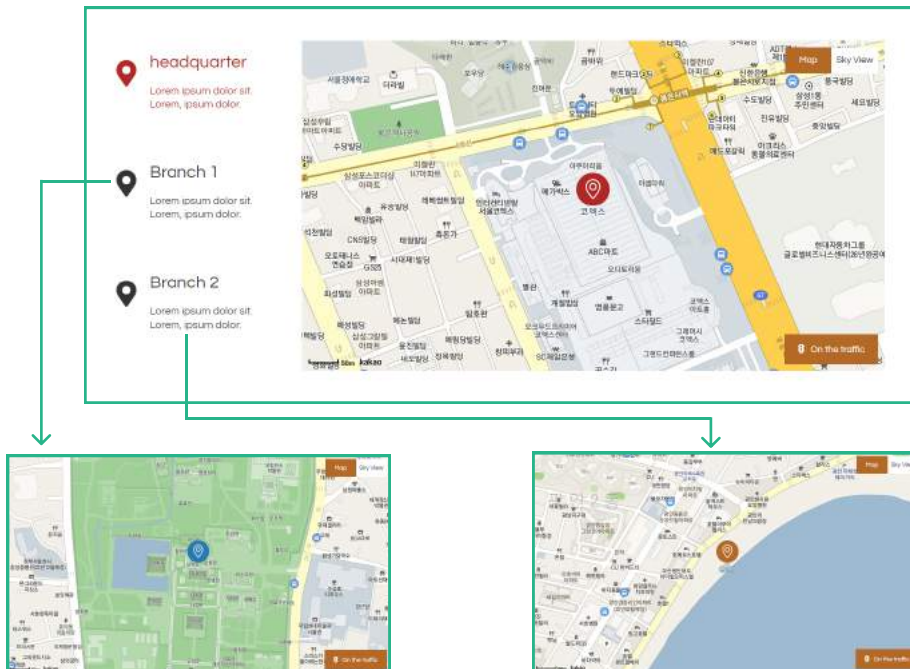
- 동영상 썸네일 클릭시 iframe으로 youtube 동영상을 팝업창으로 생성



### Our Contact

Lorem ipsum dolor sit amet consectetur adipiscing elit. Vero, aut.  
Lorem ipsum dolor sit, amet consectetur adipiscing elit.

- Address**  
12 Van Dyke St Brooklyn, NY 11231
- Phone**  
1(800)123-4567  
1(800)123-7897
- Mail**  
hello@doremi.com  
service@doremi.com
- Hours**  
Monday - Friday  
09:00am - 18:00pm



### 1. 교통정보보기



- 우측하단의 버튼 클릭시  
교통정보 보기/끄기 기능이 토글로 작동

### 2. 로드맵/위성지도



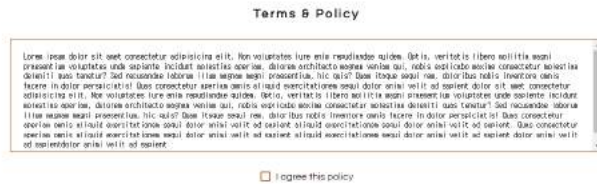
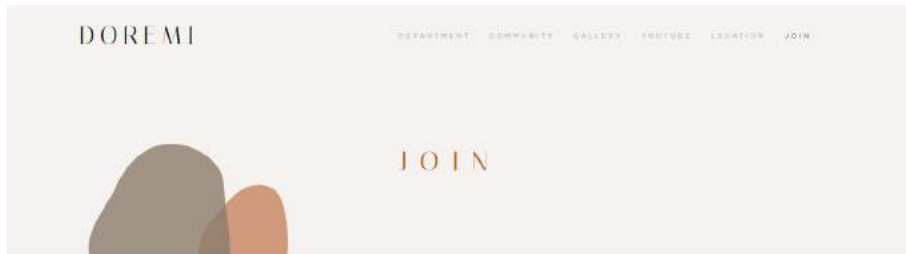
- 우측상단의 map, sky view 버튼을 통해  
지도와 위성버전보기 가능

### 3. 탭 기능

- info 배열안의 객체로 지점들의 정보를 정리하여 탭 메뉴 클릭시 순번을  
전달하여 해당 지점의 지도와 각각의 마커이미지가 호출

```
const info = [
  {
    title: "본점",
    latlng: new kakao.maps.LatLng(37.51276949548283, 127.05883923964538),
    imgSrc: process.env.PUBLIC_URL + "/img/pin.png",
    imgSize: new kakao.maps.Size(50, 50),
    imgPos: { offset: new kakao.maps.Point(25, 50) },
  },
  {
    title: "지점1",
    latlng: new kakao.maps.LatLng(37.57975318699628, 126.97709192859216),
    imgSrc: process.env.PUBLIC_URL + "/img/pin2.png",
    imgSize: new kakao.maps.Size(50, 50),
    imgPos: { offset: new kakao.maps.Point(25, 55) },
  },
  {
    title: "지점2",
    latlng: new kakao.maps.LatLng(35.15317690680852, 129.1189844468017),
    imgSrc: process.env.PUBLIC_URL + "/img/pin3.png",
    imgSize: new kakao.maps.Size(50, 50),
    imgPos: { offset: new kakao.maps.Point(25, 50) },
  },
];
```





Member Information	
USER ID	<input type="text"/>
PASSWORD	<input type="password"/>
RE-PASSWORD	<input type="password"/>
E-MAIL	<input type="text"/>
COMMENTS	<div></div>
GENDER	<input type="radio"/> MALE <input type="radio"/> FEMALE
INTERESTS	<input type="checkbox"/> Sports <input type="checkbox"/> Music <input type="checkbox"/> Game
EDUCATION	<input type="text"/>
<input type="button" value="CANCEL"/> <input type="button" value="SEND"/>	

## 1. 회원가입인증

**USER ID**

---

아이디를 입력하세요

---

**PASSWORD**

---

비밀번호를 입력하세요

---

회원가입은 이메일 아이디, 비밀번호가 특수문자 포함으로 합니다.

---

**RE-PASSWORD**

---

이름번호를 꼭 입력하세요

---

**E-MAIL**

---

이메일 주소를 입력하세요

---

이메일주소는 @를 가리지 않습니다

---

**COMMENTS**

---

---

남기는 댓글을 꼭 검토하세요.

---

**GENDER**

---

☐ MALE    ☐ FEMALE

---

성별을 선택하세요

---

**INTERESTS**

---

☐ Sports    ☐ Music    ☐ Game

---

관심사목을 하나이상 선택하세요

---

**EDUCATION**

---

---

학력을 선택해주세요

- handleChange()

input창에 변화가 생길 때마다 실행하는 함수

- `handleCheck()`  
체크박스과 라디오버튼의 체크를 확인하는 함수

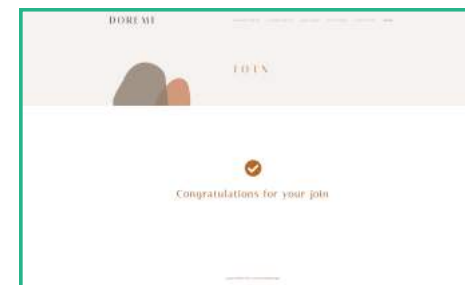
- `handleSelect()`  
셀렉트박스 옵션 선택을 확인하는 함수

- handleSubmit()

Send 버튼을 누르면 실행되는 함수

각 항목의 에러를 확인하여 에러메세지를 표출

## 2. 회원가입완료 페이지 표출



Send버튼을 클릭했을 때  
모든 항목에 에러가 없으면  
회원가입완료 페이지가 표출