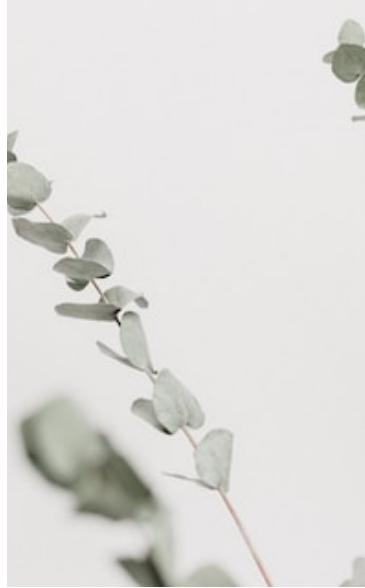




Contents

1. 프로젝트 소개	1p
2. 전체 폴더 및 파일 구조	2p
3. 반응형 웹 레이아웃	3p
4. 메인 페이지 소개	4p
5. 서브 공통 페이지 소개	5p
• 디파트먼트 페이지 소개	6p
• 커뮤니티 페이지 소개	7p
• 갤러리 페이지 소개	8p
• 유튜브 페이지 소개	9p
• 로케이션 페이지 소개	10p
• 회원가입 페이지 소개	11p



A. (에이닷)

가상의 건축 사무소 회사인 에이닷(A.)이 있다고 가정하고 홈페이지 제작

다양한 기업 홈페이지들을 검색해 보고 대부분의 기업 홈페이지가 공통적으로 가지고 있는 요소들을 컨텐츠로 채택 (조직도, 게시판, 위치안내 등)

순수 바닐라 자바스크립트를 사용하여 슬라이드, 서버통신 외 다양한 인터랙티브한 기능 제작

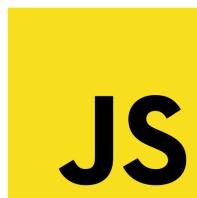
SPEC



Web Standard



SASS



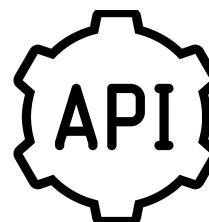
JAVASCRIPT



Responsive



Asynchronous
communication

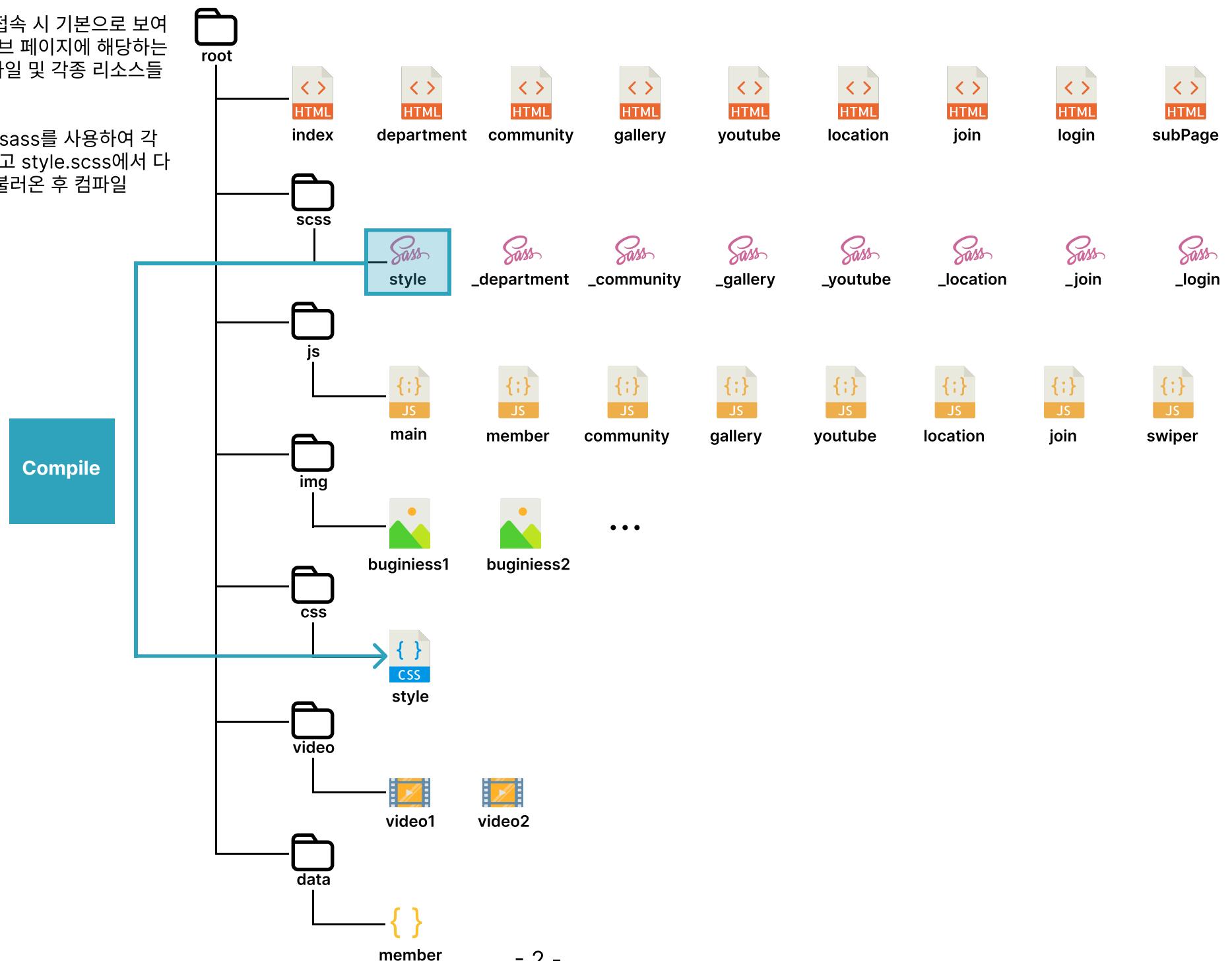


API

File Structure

루트 바로 아래에는 도메인 접속 시 기본으로 보여질 index.html 문서와 각 서브 페이지에 해당하는 문서들이 있으며, scss, js파일 및 각종 리소스들은 폴더 별로 정리하여 관리

css의 전처리기에 해당하는 sass를 사용하여 각 페이지별로 스타일을 작성하고 style.scss에서 다른 스타일들을 import하여 불러온 후 컴파일



Responsive Layout

미디어 쿼리를 사용하여 반응형 구현
분기점(break point)이 높은 순서대로 미디어 쿼리 작성(데스크탑 퍼스트)

(구)모니터 및 태블릿 가로

@media screen and (max-width: 1179px)

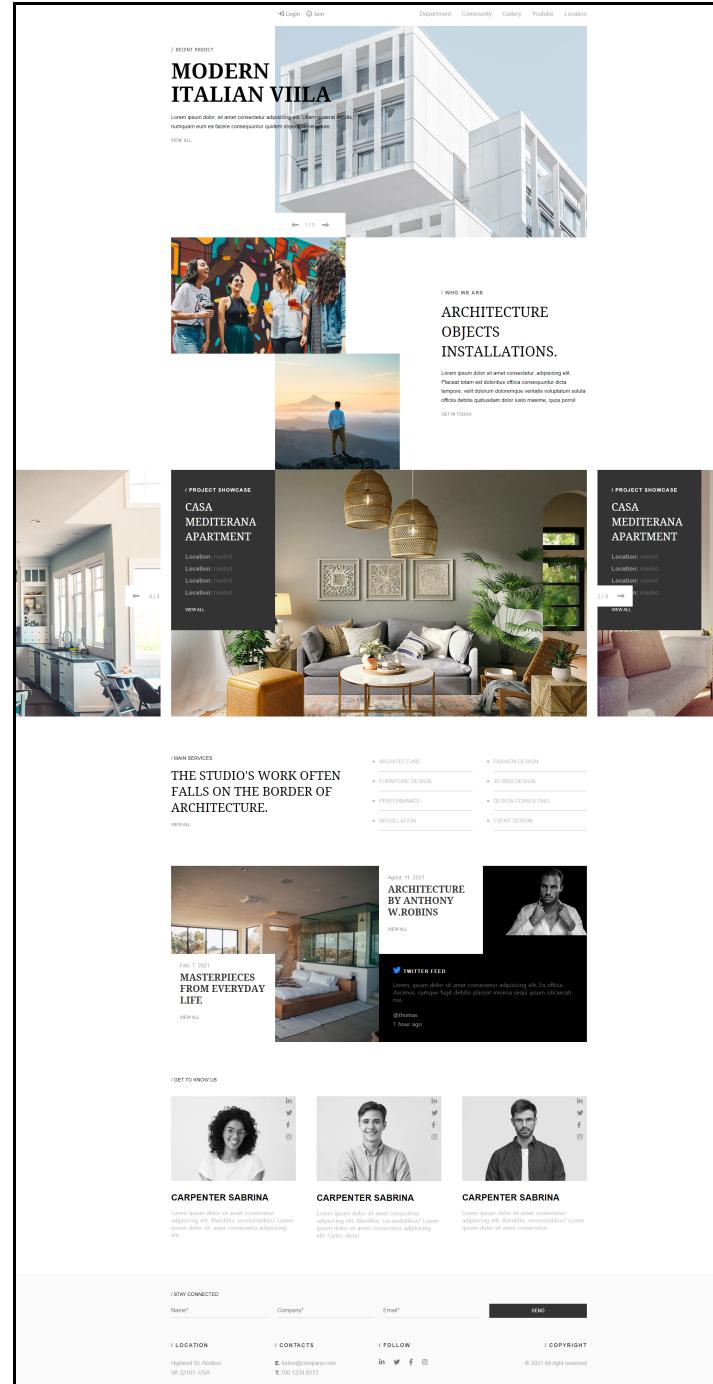
태블릿 및 모바일 가로

@media screen and (max-width: 767px)

모바일

@media screen and (max-width: 539px)

1 pc



2 tablet



3 mobile



Main page

순수 바닐라 js로 비주얼 섹션의 슬라이드를 구현하였으며, 또한 자연스러운 슬라이드를 위하여 슬라이드 애니메이션 실행 중에는 '이전', '다음' 버튼을 눌러도 이벤트 핸들러가 호출되지 않도록 최적화시킴

1 메인 페이지 구성

```
<div class="slides">
  <ul>
    <li class="active" data-index="1"></li> 페이지 구현을 위해 li 요소마다 data- 속성을
    <li data-index="2"></li> 이용하여 인덱스를 저장해 둠
    <li data-index="3"></li>
  </ul>
</div>
```

2 슬라이드 구현

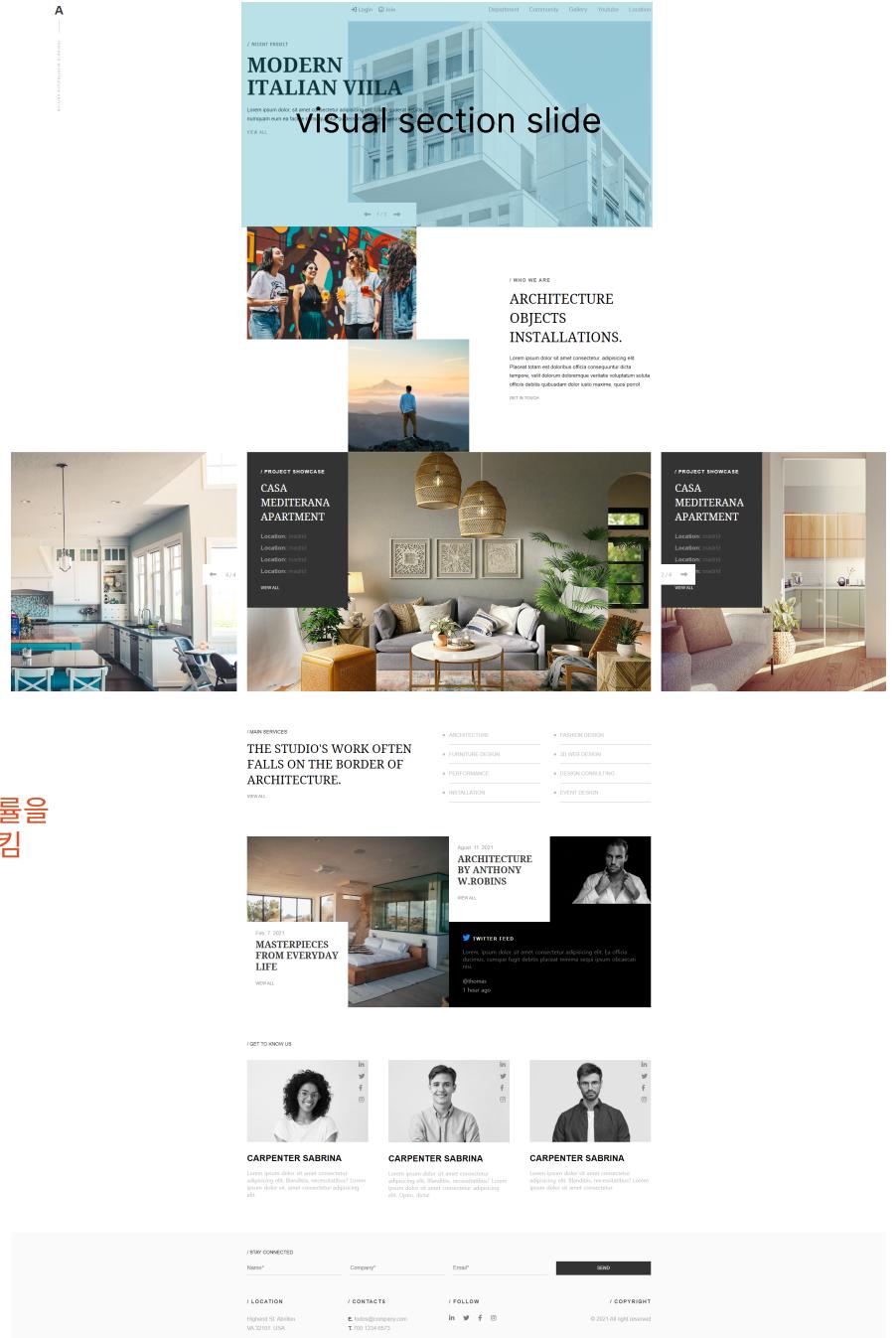
```
function makeSliding(dir, duration){ 방향('next' or 'prev')과 슬라이드 지속 시간을 인자로 전달
  let start = null;
  const currentValue = -(100 / slidesLeng);
  const goalValue = dir === 'next' ? currentValue * 2 : 0; 현재 값과 목표값은 현재의 translateX 값과,
  이동할 translateX 값 의미
```

```
return function sliding(timestamp){ requestAnimationFrame 함수를 통해 애니메이션에 사용될 함수(클로저)
  if(!start) start = timestamp;
  let progress = (performance.now() - start) / duration; 진행률
  const rafId = requestAnimationFrame(sliding);
  slides.style.transform = `translateX(${currentValue + (goalValue - currentValue) * progress}%)`;
  if( progress > 1 ){
    cancelAnimationFrame(rafId); 진행이 완료되면 애니메이션 중지
  }
}
```

3 그외 사용된 함수

initial() - 슬라이드 초기 설정 함수

throttle(callback, delay) - 자연스러운 슬라이드를 위해 슬라이드 실행(callback) 후 슬라이드 애니메이션 진행(delay)이 끝나기 전에 슬라이드가 재실행 되는 것을 방지



Subpage common parts

메인 페이지를 제외한 모든 서브페이지에 공통적으로 적용될 전체적인 틀을 미리 제작함으로써 통일성과 효율성을 높임

크게 header와 footer, 그리고 contents 섹션으로 구분되어 있으며
contents 내에는 title과 banner(비주얼 이미지)가 들어갈 위치를 미리 정의
해둠

또한 subPage 공통영역은 subPage.scss에서 따로 스타일 정의

A

modern architecture service

The diagram illustrates the common parts of a subpage template, divided into three main sections:

- Header**: Located at the top, featuring a navigation bar with links to DEPARTMENT, COMMUNITY, GALLERY, YOUTUBE, and LOCATION. A large, bold "TITLE" placeholder is positioned below the navigation.
- sub page title**: A placeholder for the specific title of the subpage, located above the "sub page banner section".
- sub page banner section**: A large, light-colored area intended for displaying visual content or promotional banners.

Footer: Located at the bottom, containing the following sections:

- / STAY CONNECTED**: Fields for Name*, Company*, Email*, and a **SEND** button.
- / LOCATION**: Address: Highend St. Abrillon, VA 32101 USA
- / CONTACTS**: Email: todos@company.com, Phone: T. 700 1234 6573
- / FOLLOW**: Placeholder for social media links.
- / COPYRIGHT**: Copyright notice: © 2021 All right reserved

Department page

fetch()로 json 데이터를 불러와 가공하여 팀원 소개 html을 동적으로 생성 및 삽입

1 디파트먼트 페이지 구성

```
<article class="introMember">
<div class="tit">
<h2>OUR TEAM</h2>
<p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni
    voluptates beatae est molestias deleniti quia, earum, facere
    temporibus quibusdam assumenda quasi quas nulla, libero
</p>
</div>
<div class="role"> js로부터 팀원 소개 html 동적 삽입
</div>
</article>
```

2 json 데이터 네트워크 통신

```
const role = document.querySelector('.role');

fetch('data/members.json') json 데이터를 get 요청
.then(data=>{
  return data.json(); json 데이터 객체로 변환
})
.then(json=>{
  const profiles = json.data;
  let tags = "";

  profiles.map(profile=>{
```

```
    tags += `
      <article class="profile">
        <div class="pic">
          <img src=${profile.pic} alt="">
        </div>
        <strong>${profile.name}</strong>
        <p>${profile.position}</p>
      </article>
    `)
  role.innerHTML = tags; 생성된 article을 html에 동적 삽입
})
```

팀원들의 정보(이름, 사진 etc..)가 담겨있는 배열을 순환하면서 팀원 개개인을 소개하는 article 생성

A

DEPARTMENT

OUR ARCHITECTURE STORY AND HIDDEN ROLE

CEO

Iggy Azalea

Her Works

OUR TEAM

ARCHITECT

Sabrina C.

Geirge B.

Tom T.

Jane M.

Richard G.

Gregor H.

Mike A.

Anna L.

Community page

탭 메뉴를 구현하여 메뉴 별로 table 태그를 사용한 게시판과 ul, li 태그를 사용한 faq 툴을 구현

1 커뮤니티 페이지 구성 (Tab)

```
<dl id="tab">
<dt class="on">
  <button href="#">
    <svg>....</svg>
    PRODUCTS
  </button>
</dt>
<dd class="on">
  <ul id="faq">
    <li>
      <a href="#">
        <span>Products</span>
        <p>기름 얼룩이 있어요.</p>
      </a>
      <div class="contents">...</div>
    </li>
    <ul>
      <li>...</li>
    </ul>
  </ul>
</dd>
</dl>
```

dt, dd를 이용해 탭 메뉴 구성

notice 탭의 컨텐츠는 테이블로 구성되어 있으며 나머지 탭의 컨텐츠는 faq 툴을 구성

2 탭 메뉴 svg 애니메이션

```
@keyframes strokeNote {
  0% {stroke-dashoffset: 228}
  100% {stroke-dashoffset: 0}
}
```

스트로크의 시작점을 의미하는 dashoffset 속성을 스트로크의 전체 길이로 설정하여 보이지 않게 한 후 0으로 만들어 줘 마치 선이 그려지는 효과를 만듬

The screenshot shows a community page with a header "FAQ" and a sub-header "How can we help you? Choose one of the categories below". Below this are four icons: NOTICE (document), PRODUCTS (gift box), BUILDING (building), and COST (database). The main content area has a "Total 10" heading followed by a list of 10 questions under the "Product" category. The 10th question is highlighted with a blue background and the text "click event". A large green arrow points downwards from the "click event" text towards a pagination bar at the bottom.

Category	Question
Product	기름 얼룩이 있어요.
Product	새 제품에서 냄새가 나요.
Product	가죽제품 관리는 어떻게 해야 하나요?
Product	제품을 받았는데, 부분마다 착색감이 다른 것 같아요.
Product	기존 구매 제품과 새로 구매한 제품과의 색상차이가 있어요.
Product	실제 제품과 모니터 상의 제품 색상이 달라요.
Product	제품이 구겨져 있어요.
Product	사용한 지 얼마 안되었는데 주름이 졌어요.
Product	제품 부분 별 색상이 달라요.
A	Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe sapiente molestias, nam, aspernatur veritatis similique necessitatibus aliquid asperiores, quia mollitia neque! Soluta amet excepturi deleniti fugit sunt error eius libero.

```
faqs.forEach(faq => {
  faq.addEventListener('click', e => {
    e.preventDefault();
    const liElem = e.target.closest('li');
    if(liElem.classList.contains('on')){
      liElem.classList.remove('on')
    }else{
      for(let li of [...e.currentTarget.children]){
        li.classList.remove('on');
      }
      liElem.classList.add('on');
    }
  })
})
```

faq 리스트 클릭 시 해당 리스트에 'on' 클래스가 있으면 제거하고 그렇지 않으면 모든 리스트의 'on' 클래스를 제거하고 클릭이 발생한 해당 리스트에만 'on' 클래스 추가

Gallery page

flickr API 통신을 통해 flickr 서버로부터 이미지 데이터를 제공 받아 이미지 갤러리 구성.

키워드를 입력하면 해당 키워드에 해당하는 이미지들로 갤러리를 구성할 수 있는 input 구성

1 갤러리 페이지 구성

```
<div class="sub_content">
  <div class="inner">
    
    <div class="searchBox">
      <input type="text" id="search" placeholder="Insert keywords">
      <button class="btnSearch">Search</button>
    </div>
    <div class="gallery">
      flickr API 통신을 통한 동적 이미지 삽입
    </div>
  </div>
</div>
```

2 flickr API 통신

```
const url = `${base}method=${method}&api_key=${api_key}
&per_page=${per_page}&format=${format}&nojsoncallback=1`;
```

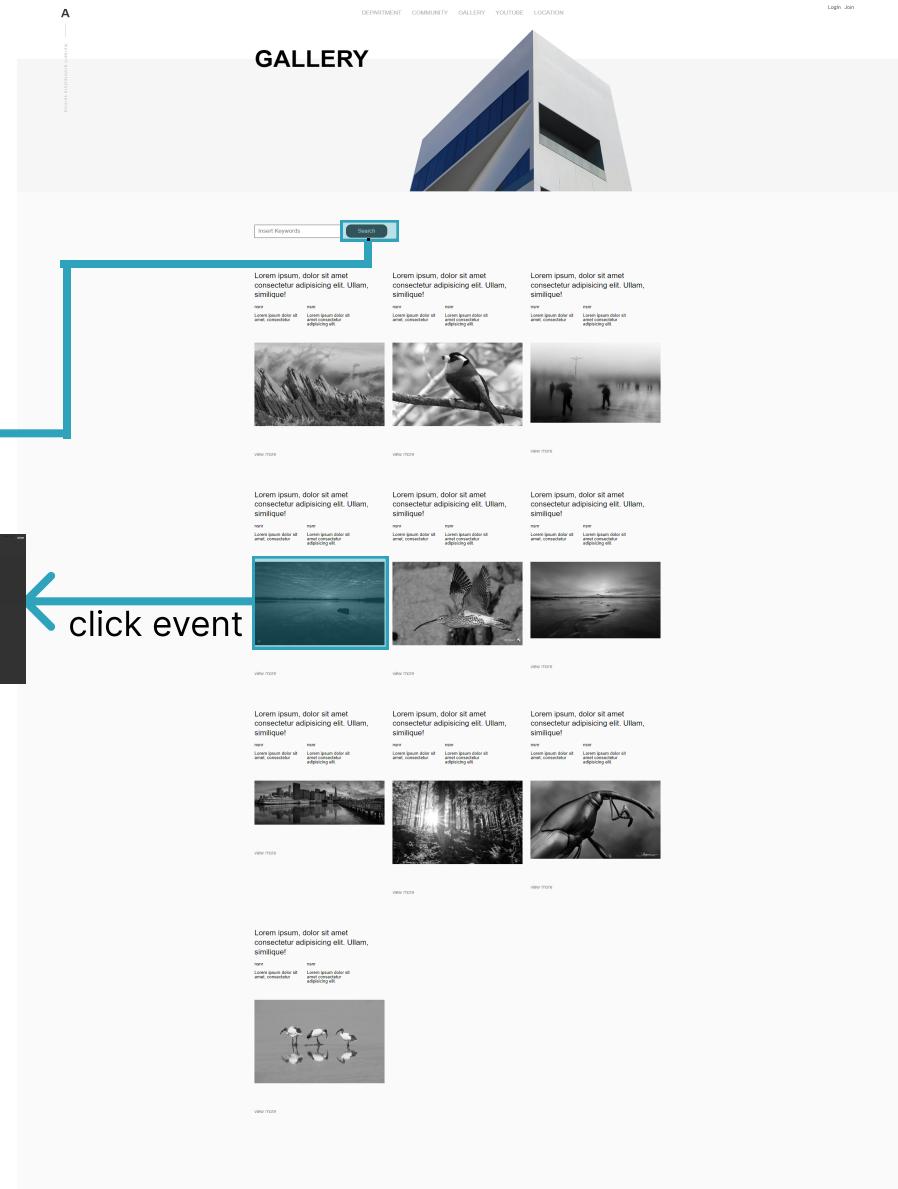
```
function callPhoto(url){
  fetch(url)
  .then(data=>data.json())
  .then(json=>{
    getPhotoInfo(json);
  })
}
```

3 그 외 사용된 함수

getPhotoSize() - callPhoto에 의해 응답받은 이미지들의 다양한 사이즈들이 담긴 객체를 가져오고
사이즈 객체와 함께 createList 함수 호출

createList() - 이미지 아티클 생성

delayLoading() - flickr api로부터 이미지가 로드 될 때까지 gif (로딩)이미지를 on 시키고 이미지 로드가 완료되면 off시킴



Youtube page

Youtube api를 통해 유튜브 채널에 있는 플레이 리스트와 동영상을 불러와
유튜브 페이지를 구성

1 유튜브 페이지 구성

비디오 섹션 별 다른 재생목록의 동영상 삽입

```
<section class="video videoSection2 wrap">
<div class="txt">
<span>/ ARCHITECTURE</span>
<h3>Various Projects</h3>
<p>
<span>
we have ventured into a variety of new projects in recent years.
</span>
The YouTube on the right side will show you a variety of our
projects around the world
</p>
</div>
<article></article>
<article></article>
<article></article>
</section>
```

JS로 유튜브 썸네일 및 제목, 설명 동적 삽입



pop up

2 youtube API 통신

```
function callData(playlistId, container){
const url = `https://www.googleapis.com/youtube/v3/playlistItems?part=${part}
&key=${key}&playlistId=${playlistId}`; 인자로 전달받은 playList에 해당되는 동영상 목록 요청
```

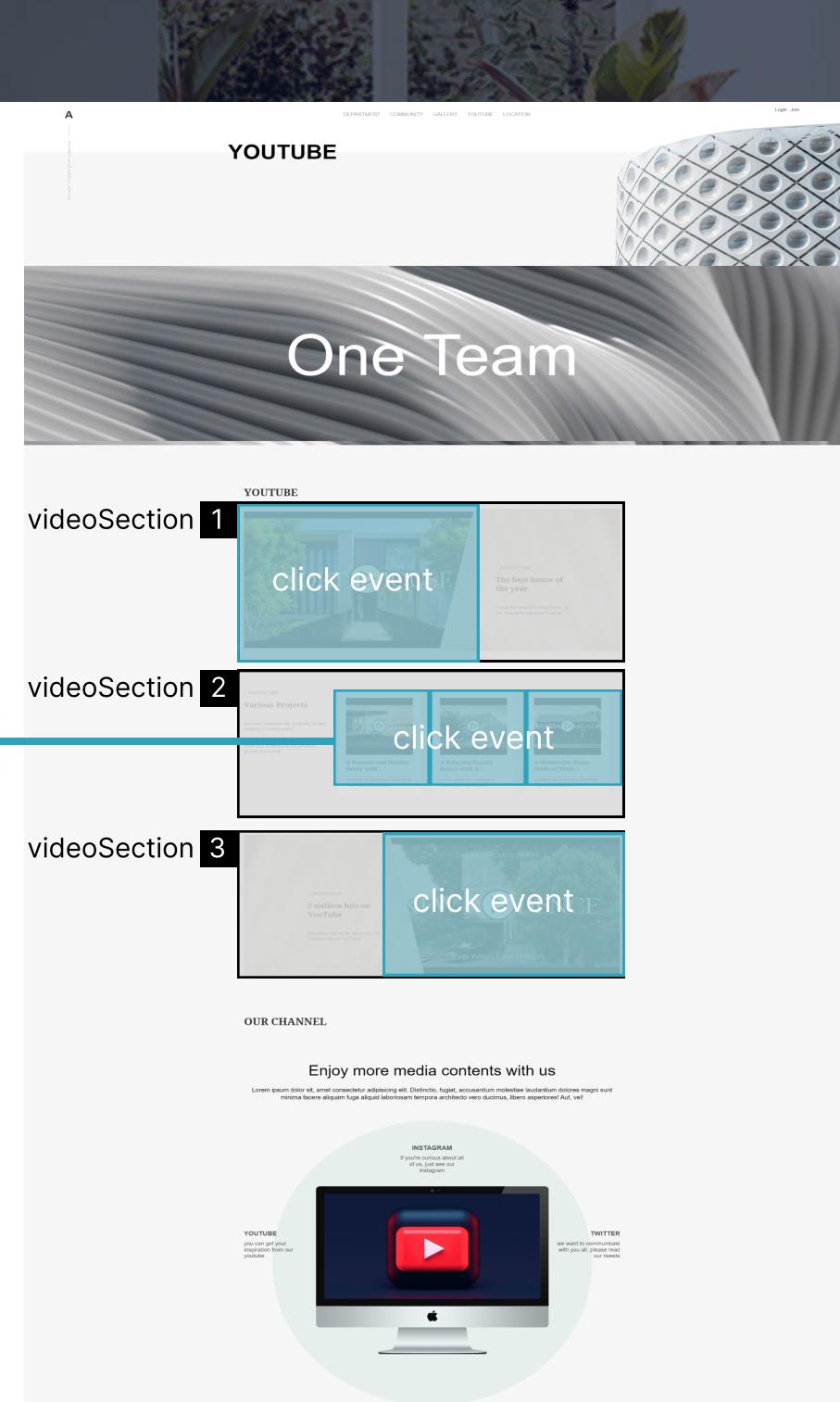
```
fetch(url)
.then(response=>response.json())
.then(json=>{
const items = json.items;
items.forEach((item, idx)=>{
  :
})
```

브라우저 내장함수인 fetch()로 API 통신 후, 응답받은 동영상
목록을 가공하여 인자로 전달받은 videoSection 내 아티클
(썸네일, 제목, 설명) 생성

3 그 외 사용된 함수

callPlayLists(channelId) - Youtube api를 사용하여 인자로 전달받은 채널의 재생목록 정보를
불러온 후 callData 호출

textSlice(txt, length) - 동영상 제목 및 설명의 길이가 length 이상이면 자르고, 말줄임표(...) 추가



Location page

Youtube api를 통해 유튜브 채널에 있는 플레이 리스트와 동영상을 불러와
유튜브 페이지를 구성

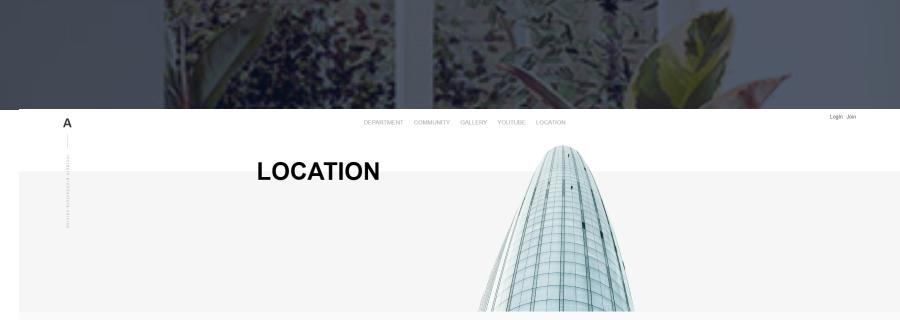
1 로케이션 페이지 구성

```
<div id="map"></div> 지도가 들어갈 영역
<div class="zoomControl">
  <span class="zoom_in" onclick="zoomIn()>
    <i class="icon_plus" aria-hidden="true"></i>
    <em class="hidden">확대</em>
  </span>
  <span class="zoom_out" onclick="zoomOut()>
    <i class="icon_minus" aria-hidden="true"></i>
    <em class="hidden">축소</em>
  </span>
</div>
```

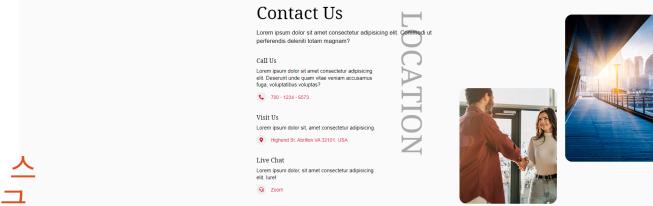
지도에 줄인, 줄이웃 버튼



aria-hidden 속성을 true로 설정하여 폰트어썸 유니코드를 스
크린 리더기가 읽지 못하게 하고 대체 텍스트를 넣어주어 접근
성을 향상



LOCATION

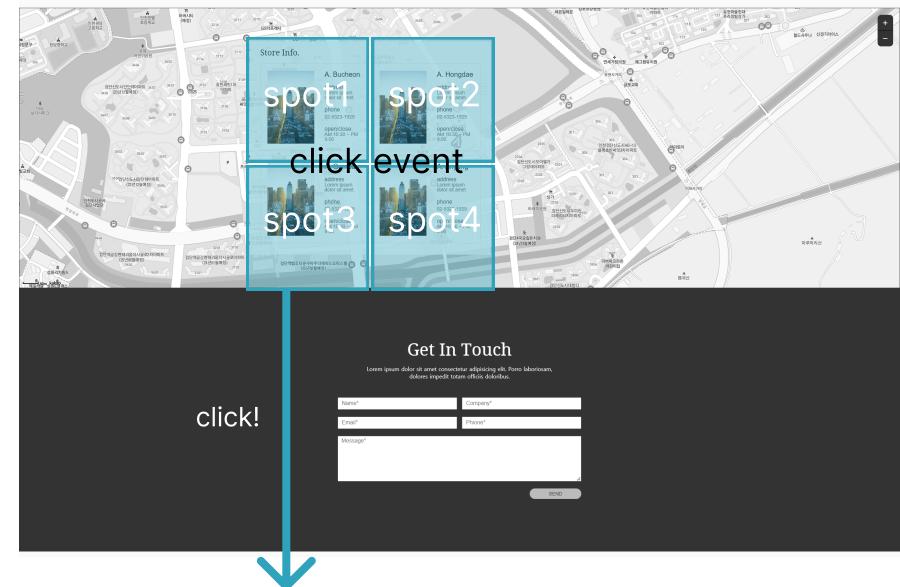


2 카카오 맵 지도 생성

```
const mapContainer = document.getElementById('map');
const mapOptions = {
  center: new kakao.maps.LatLng(37.5048632, 126.7633616),
  level: 3,
};
```

```
const map = new kakao.maps.Map(mapContainer, mapOptions);
```

맵이 들어갈 박스와 옵션을 인자로 전달하여
지도 생성



```
spots.forEach((spot, index) =>{
  spot.addEventListener('click', e =>{
    const moveLatLon = markerPositions[index].latlng;
    map.setCenter(moveLatLon);
  })
})
```

각 지점들을 순회하면서 클릭 이벤트 추가
해당 지점의 정보가 담겨 있는 markerPosition에서 좌표정보
를 가져온 후 해당 좌표로 지도 중심좌표를 이동시킴

3 그 외 사용된 함수

setZoomable(zoomable) - 맵의 줌 가능 여부 설정

zoomIn() - 지도를 줌 인 시켜 상세하게 보여줌

zoomOut() - 지도를 줌 아웃 시켜 전체적으로 보여줌

4 지도 마커 정보

```
const markerPositions = [
  {
    title: '부천점',
    latlng: new kakao.maps.LatLng(37.5048632, 126.7633616),
  },
  {
    title: '홍대점',
    latlng: new kakao.maps.LatLng(37.5614033, 126.9196564),
  },
  :
]
```

하단의 확인 버튼 클릭 시 유효성 검사를 통해 값의 유효성 여부를 판단하고 그에 따른 결과값을 보여주는 회원가입 폼으로 구성

1 회원가입 페이지 구성

```
<form action="#" method="get" id="insertInfo">
<fieldset>
<legend class="hidden">기본정보 입력</legend>
<table>
<caption>BASIC INFO.</caption> 웹 접근성을 위해 테이블의 내용을 집약적으로 알 수 있는
caption 삽입
<tr>
<th scope="row"><label for="name">NAME*</label></th>
<td>
<input type="text" id="name" name="name"
placeholder="Insert Your Name" required>
</td>
</tr>
</table>
</fieldset>
</form>
```

화면 낭독기는 왼쪽부터 오른쪽으로 순차적으로 표를 읽어 주기 때문에 화면 낭독기에 의존하는 사용자들은 테이블 정보를 이해하기 어려움

제목 셀에 scope 속성을 줌으로써 화면 낭독기가 scope 값에 따라 셀을 읽는 순서를 달리 하도록 해 접근성을 향상시킴

2 유효성 검사

```
function isTxt(name, len){  input의 name 속성 값과 설정하기 원하는 최소 length를 인자로 전달
if(len === undefined) len = 5;
let input = form.querySelector(`[name=${name}]`);
let txt = input.value;

if(txt.length >= len){
  const errMsgs = input.closest("td").querySelectorAll("p");
  if(errMsgs.length>0) input.closest("td").querySelector("p").remove();
  return true;  유효성 충족 시 true 리턴
}else{
  const errMsgs = input.closest("td").querySelectorAll("p");
  if(errMsgs.length>0) input.closest("td").querySelector("p").remove();

  const errMsg = document.createElement("p");
  errMsg.append(`해당항목을 ${len}글자 이상 입력하세요`);
  input.closest("td").append(errMsg);
  return false;
}  유효성 불충족 시 에러 메시지를 표시하고 false 리턴
```

유효성 검사 함수들을 실행하고 모두 충족 시 즉, 모든 함수가 true를 반환할 시에만 페이지를 새로고침

USERS AGREEMENT.

Lorem ipsum dolor sit amet.!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam error corporis culpa saepe nesciunt in ea repellendus dolor minima officiis!

Lorem, ipsum.

Lorem ipsum dolor sit amet consectetur adipisicing.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni cumque ullam alias animi dolore ex recusandae omnis soluta provident doloribus. Beatae sit corrupti facere eos consequuntur, repudiandae rem illo vitae tenetur provident? Id obcaecati

Do you agree with the users agreement?

Yes No

BASIC INFO.

NAME*	<input type="text" value="Insert Your Name"/>
USER ID*	<input type="text" value="Insert Your Id"/>
PASSWORD*	<input type="password" value="Insert Your Password"/>
RE-PASSWORD*	<input type="password" value="Reinsert Your Password"/>
EMAIL*	<input type="text" value="Insert Your Email"/>
PHONE*	<input type="text" value="Insert Your Phone Number"/>

EVENT INFO
RECEPTION *

E-MAIL PHONE MAIL SMS MESSAGE

cancel

confirm

click event



```
btnSubmit.addEventListener('click', e=>{
  if(!isAgree("agreement")) e.preventDefault();
  if(!isTxt("name", 2)) e.preventDefault();
  if(!isTxt("id", 5)) e.preventDefault();
  if(!isTxt("phone", 11)) e.preventDefault();
  if(!isPwd("pwd1", "pwd2", 8)) e.preventDefault();
  if(!isEmail("email")) e.preventDefault();
  if(!isCheck("reception")) e.preventDefault();
})
```