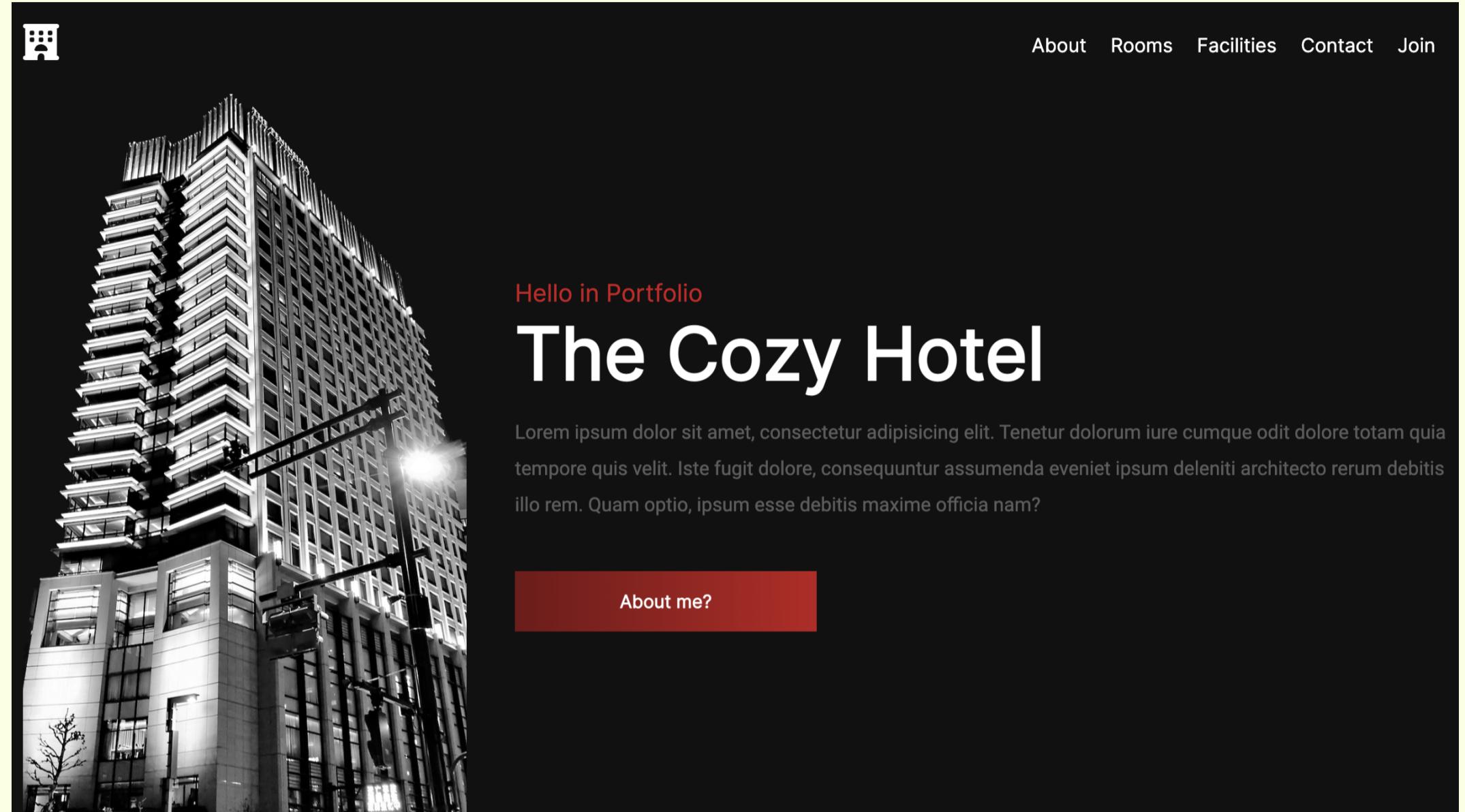


React Portfolio

The Cozy라는
가상의 호텔기업의 사이트를 상상하여 제작하였습니다.

컬러디자인은 고급스러운 느낌을 주기위해
어두운색의 배경을 선택하였으며,

강조색으로는 빨간색을 사용하여 대비효과를 주어
중요 포인트에 집중도를 높혔습니다.



반응형 웹페이지

다양한 환경에서 웹페이지
구동을 가능하게 하기위해
데스크톱, 태블릿, 모바일
버전의 반응형 웹페이지를 구
현했습니다.

Redux-toolkit

기존의 단방향 데이터이동이
아닌 양방향 데이터이동과
향후 유지보수의 용이성을 위
하여 **Redux**를 사용했습니다.

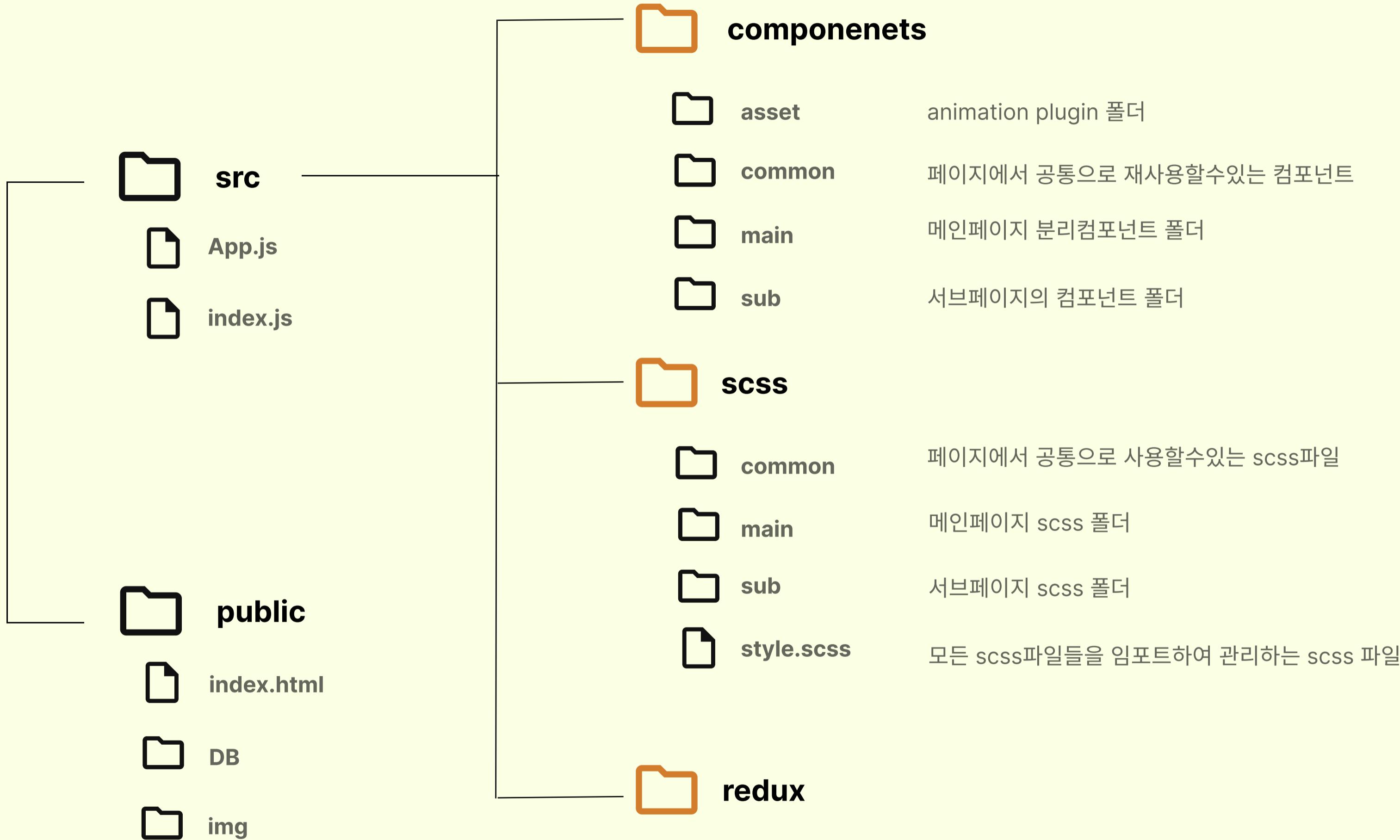
API사용

Kakao Map, Youtube,
Flicker의 **API**를
불러와 사용하였습니다.

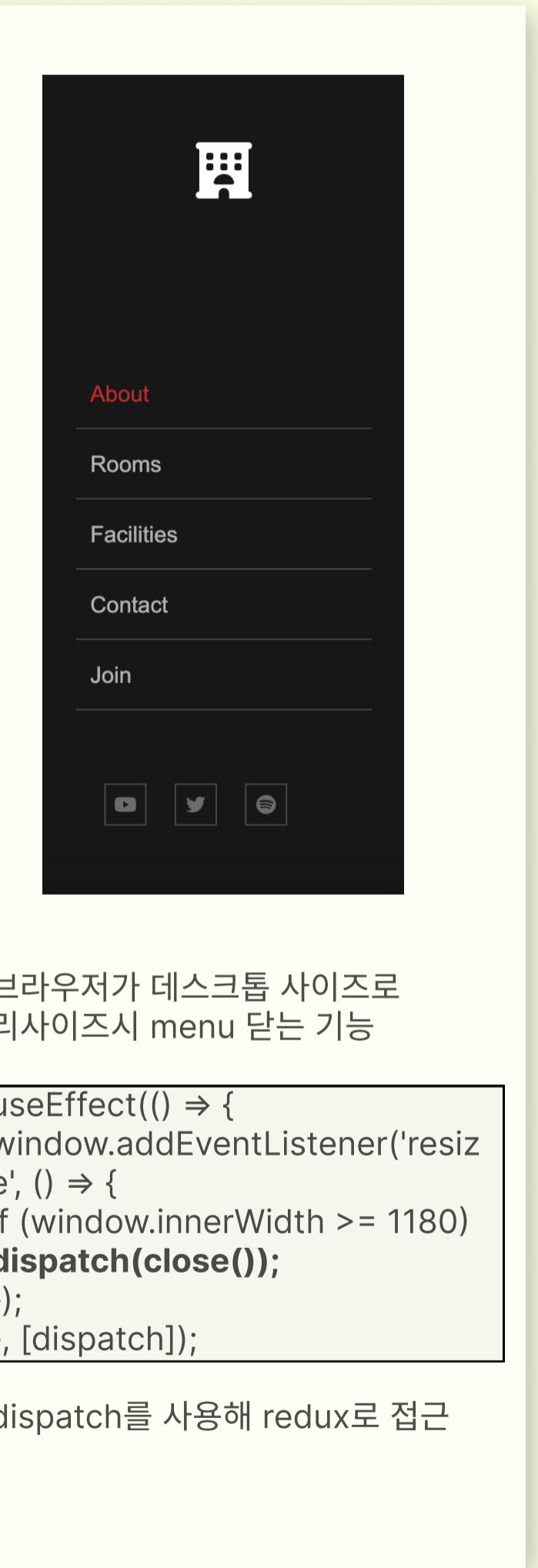
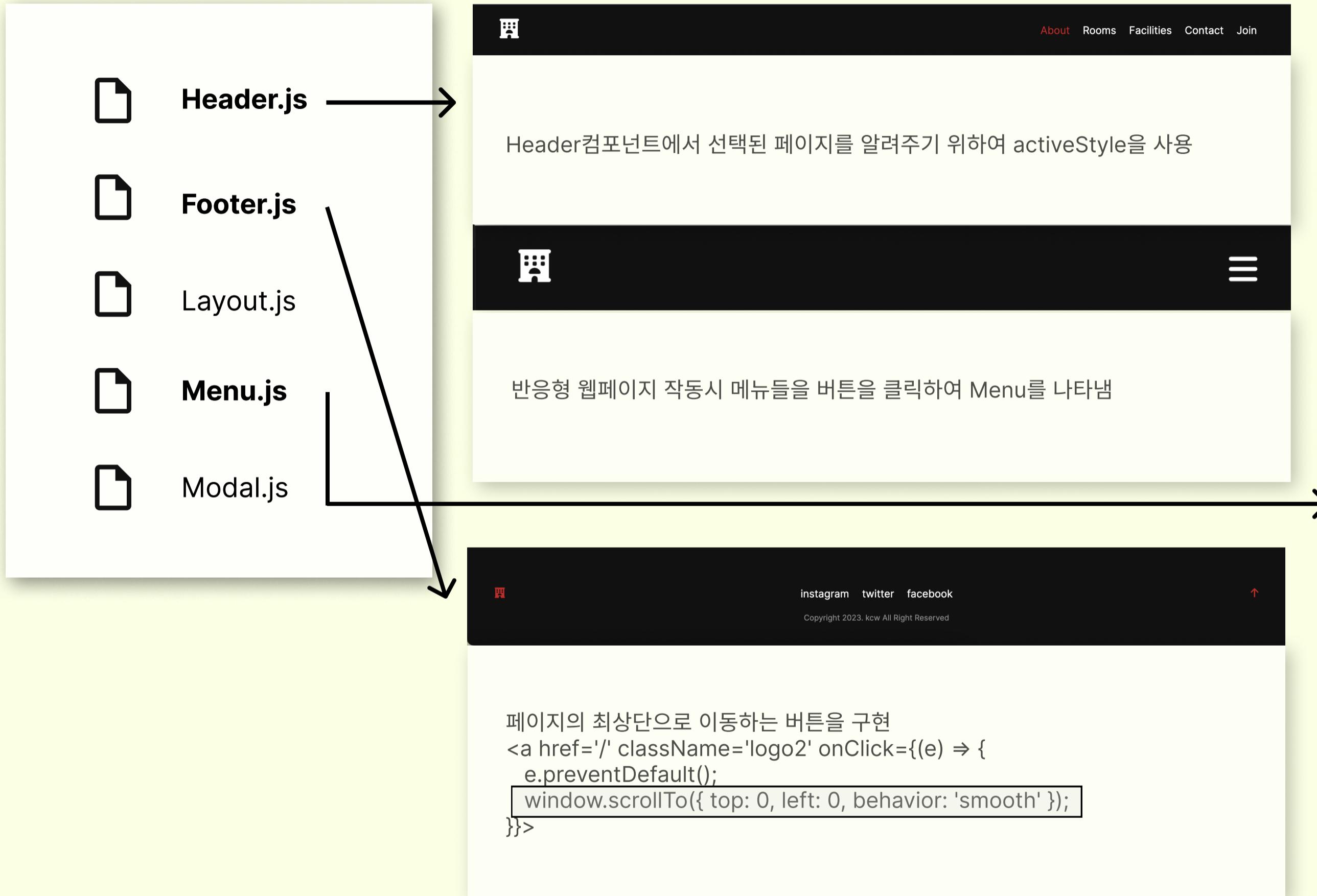
CRUD 기능

Contact에 **CRUD**기능을 구
현했으며, 저장된 글을
LocalStorage에 담아 메인
페이지에서도 볼수있습니다.

Directory



components - common



components - common

- Header.js
- Footer.js
- Layout.js
- Menu.js
- Modal.js

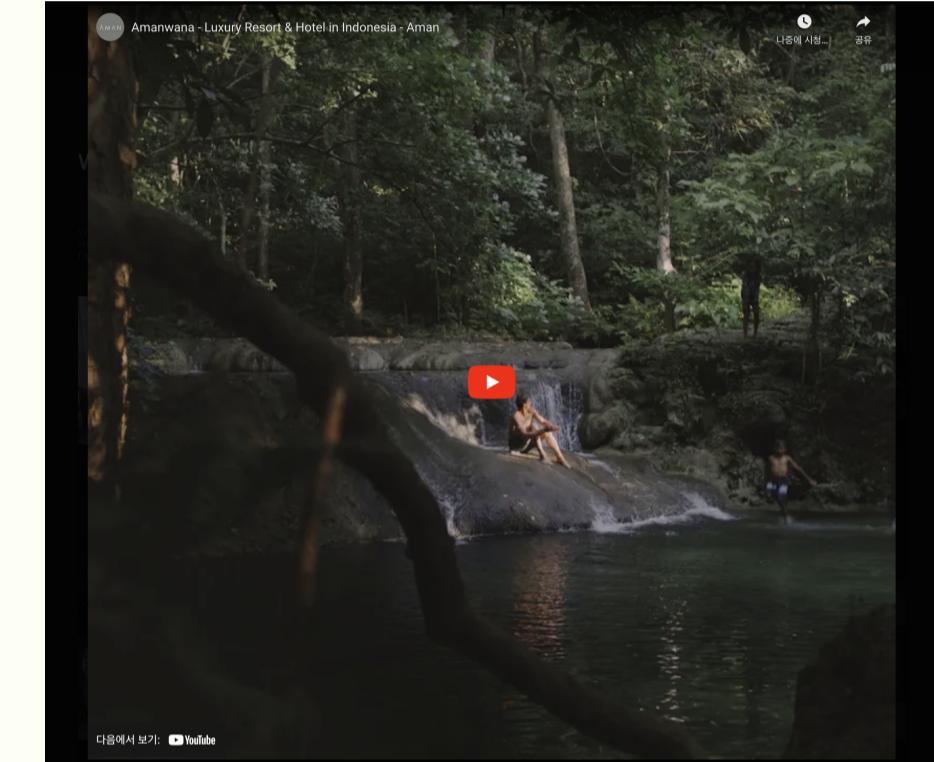
forwardRef를 사용하여
props, 와 ref를 넘겨줘
내부 html에 접근

```
const Modal = forwardRef((props, ref) => {  
  const [Open, setOpen] = useState(false);
```

Modal을 여는 함수를 부모 컴포넌트로 전달

```
useImperativeHandle(ref, () => {  
  return { setOpen: () => setOpen(true) };  
});
```

Video.js



Facilities.js

```
<Modal ref={open}>  
<iframe  
title={vids[index]?.id}  
src={`https://  
www.youtube.com/embed/  
${vids[index]?.snippet.resource  
Id.videoid}`}  
></iframe>  
</Modal>
```

```
<Modal ref={open}>  
<img  
src={`https://  
live.staticflickr.com/  
${Items[Index]?.server}/  
${Items[Index]?.id}  
_${Items[Index]?.secret}  
_b.jpg`}  
alt={Items[Index]?.title}  
>  
</Modal>
```

Main 페이지

- Visual.js
- Top.js
- Rank.js
- Gallery.js
- Vids.js
- Banner.js
- Pics.js
- News.js
- Faq.js

LOOK AROUND

Enjoy the various attractions and natural beauty that you can experience during your stay at the hotel

Hilton Bali Resort | Hotel Vid

Renaissance Downtown Hotel Dub

CONTINENTAL HOTEL Budapest pro

redux 스토어에 담아 놓았던 데이터를 불러온후 .map함수를 통하여 페이지를 구현

```
const pics = useSelector((store) =>  
  store.flickr.data);  
  
const Vids = useSelector((store) =>  
  store.video.data);
```

COMMUNITY

Check-in time

I am arriving early in the morning. Can I check in before 3 pm?

Breakfast menu

I have some dietary restrictions. What kind of food do you serve for breakfast?

Parking fee

I am planning to rent a car during my stay. How much is the parking fee per day and where can I park?

Room service

I want to order some food and drinks to my room. Do you offer room service 24/7 and what is the menu?

초기 Dummy 데이터들을 localStorage에 담기

```
const getLocalData = () => {  
  const dummies = [...(더미데이터)];  
  const data = localStorage.getItem('post');
```

이후 작성된 데이터가 있으면 작성된 데이터를 받아오고 데이터가 없으면 더미데이터를 사용

```
if (data) return JSON.parse(data);  
else return dummies;
```

useEffect로 데이터값 가져오기

```
const [Posts] = useState(getLocalData());  
  
useEffect(() => {  
  localStorage.setItem('post',  
    JSON.stringify(Posts));  
}, [Posts]);
```

About 페이지



About.js

기업의 조직도와
호텔을 나타내는
유튜브 비디오를
소개하는 서브페이지



member.json

```
"members": [  
  {  
    "name": "Ashlyn",  
    "position": "General Manager",  
    "pic": "dep1.jpg"  
  },  
  {  
    "name": "Larry",  
    "position": "Beverage Manager",  
    "pic": "dep2.jpg"  
  },  
  ....
```

PUBLIC/DB폴더에
직원들의 정보가 담긴
json파일을 만들어서
사용
→ 나중에 수정할때 용이

```
useEffect(() => {  
  axios.get(`${process.env.PUBLIC_URL}/DB/member.json`)  
    .then((json) => {  
      setMember(json.data.members);  
    });  
});
```

Team COZY를 소개합니다.
Je vous présente Team COZY.
Let me introduce Team COZY.
Team COZYを紹介します。
介绍一下Team COZY。



Ashlyn
General Manager



Larry
Beverage Manager



Tyshawn
Food Manager



Celeste
Executive Housekeeper



This page introduces the hotel executives. The executives are at the forefront of the hotel industry with their expertise and experience. The executives operate and manage various hotels around the world, and strive to provide the best service and comfort to customers. The executives play a big role in shaping the future of the hotel industry. If you want to know more about them, please click on their profiles below.

axios를 통하여
json데이터를 가져
올때 환경변수인
process.env.PUBLIC_URL을 사용
→ 보안성을 위하여

About 페이지 - 비디오



About.js

```
const vids = useSelector(store)
  → store.video.data)
```

메인페이지의
유튜브를 불러오는
방식과 같은방식으로
store에 담겨져있던
데이터를 가져온다

```
onClick={() => {
  open.current.setOpen();
...
<Modal ref={open}>
```



Modal.js

```
useImperativeHandle(r
ef, () => {
  return { setOpen: () =>
    setOpen(true);
  });
});
```

Modal 컴포넌트에
useRef를 사용한뒤
setOpen로 모달창을 띄움



redux/videoSlice.js

```
const youtubeSlice = createSlice({
  name: 'video',
  initialState: {
    data: [],
    isLoading: false,
  },
}
```

```
extraReducers: {
  [fetchVideo.pending]: (state) => {
    state.isLoading = true;
  },
  데이터 요청 시작
},
```

```
[fetchVideo.fulfilled]: (state, action) => {
  state.isLoading = false;
  state.data = action.payload;
},
```

데이터 요청 성공

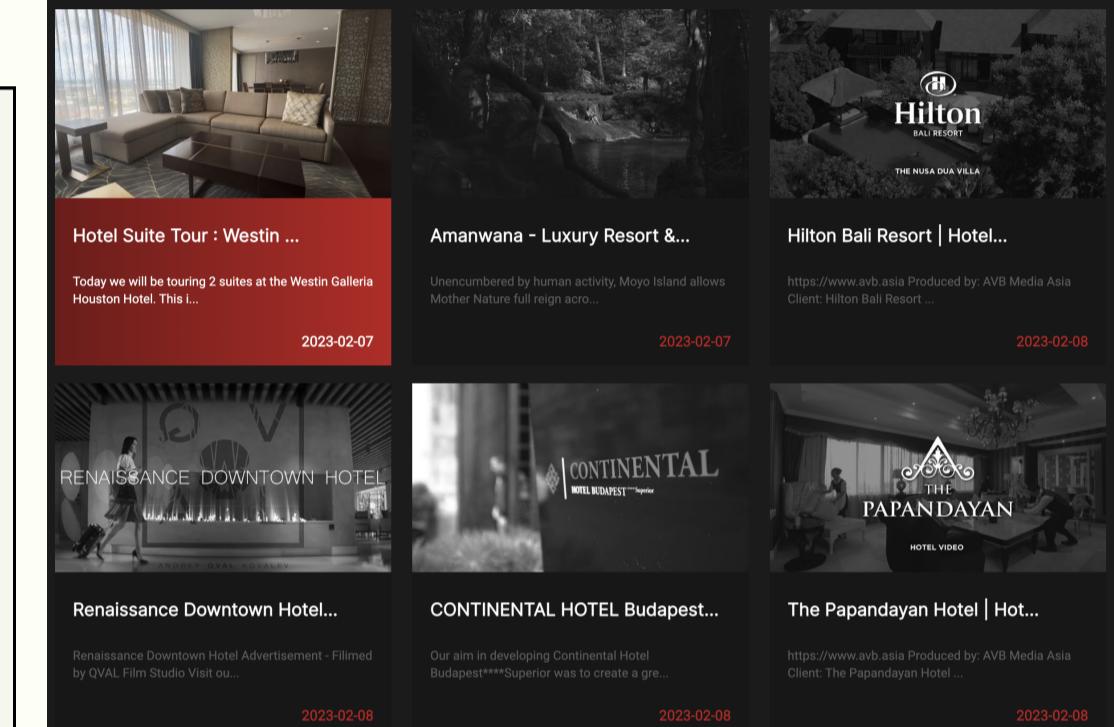
```
[fetchVideo.rejected]: (state, action) => {
  state.isLoading = false;
  state.data = action.payload;
},
```

데이터 요청 실패

```
},
```

```
});
```

VIDEO



```
export const fetchVideo =
  createAsyncThunk('video/requestVideo', async
() => {
```

```
const url = `https://www.googleapis.com/
youtube/v3/playlistItems?
part=snippet&key=${key}
&playlistId=${playlistId}&maxResults=${num}`;
```

```
const response = await axios.get(url);
return response.data.items;
});
```

비동기로 api값을 불러온뒤
스토어에 저장하여 활용

Rooms 페이지

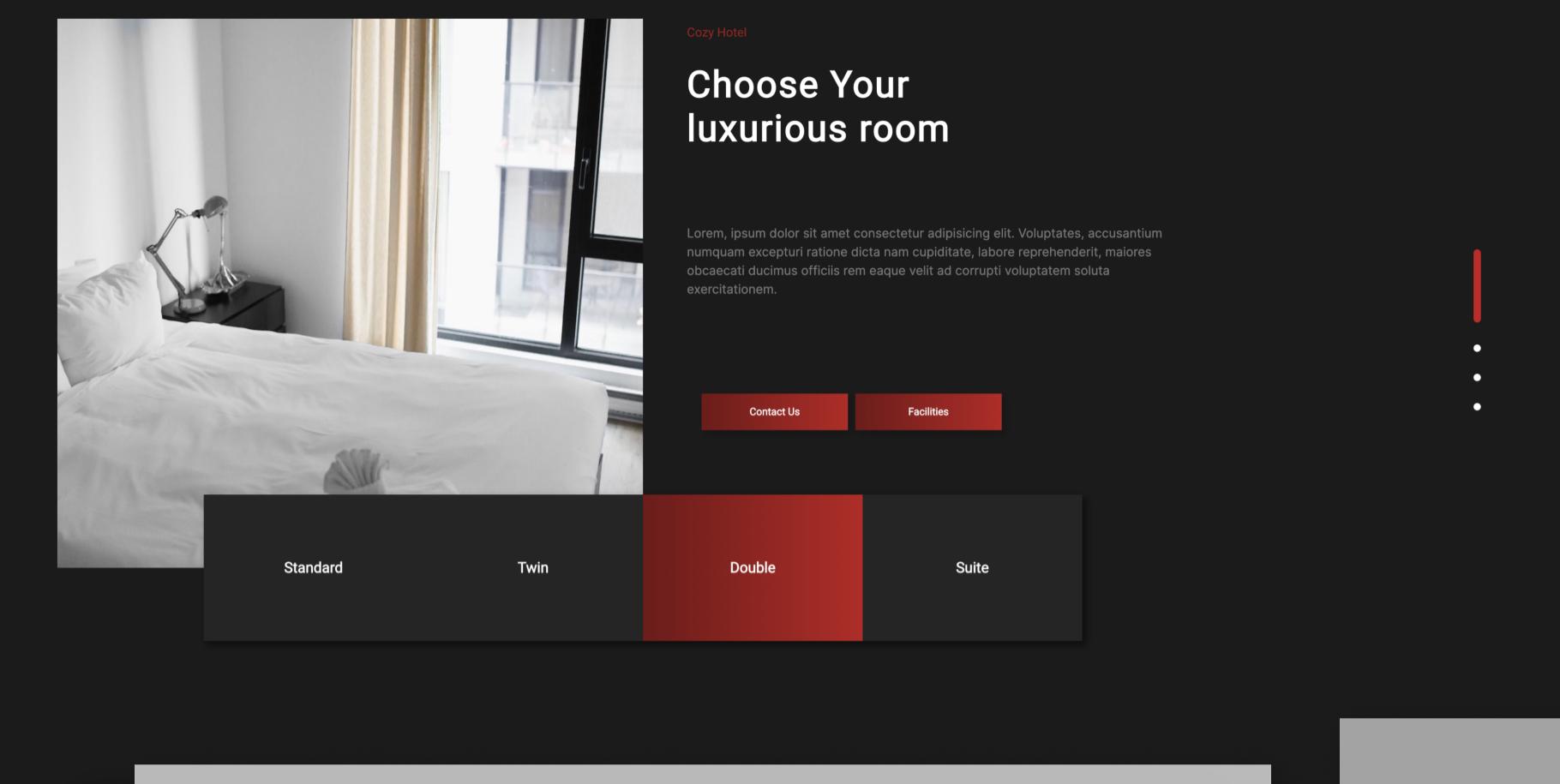


Btns.js

한 페이지를 구성하고 있는 여러 섹션의 최상단 Y값인 offsetTop을 구하여서 각 페이지로 이동, 각 페이지에 도달할 시 css애니메이션 값을 활성화 하도록 만드는 컴포넌트

```
const getPos = useCallback(() => {
  pos.current = []; 리사이즈시 기준의 배열값 초기화
  const secs =
    btnRef.current.parentElement.querySelectorAll('.myS
  scroll');
  for (const sec of secs)
    pos.current.push(sec.offsetTop);
  setPos(pos.current);
}, [setPos]);
```

현재위치를 기준으로 각 섹션의 offsetTop값을 구하기 위한 함수



```
const activation = useCallback(() => {
  const btns = btnRef.current.children;
  const scroll = window.scrollY;
  const base = -window.innerHeight / 3;
  const secs =
    btnRef.current.parentElement.querySelectorAll('.myS
  scroll');
  setScrolled(scroll);

  pos.current.forEach((pos, idx) => {
    if (scroll >= pos + base) {
      for (const btn of btns) btn.classList.remove('on');
      for (const sec of secs) sec.classList.remove('on');

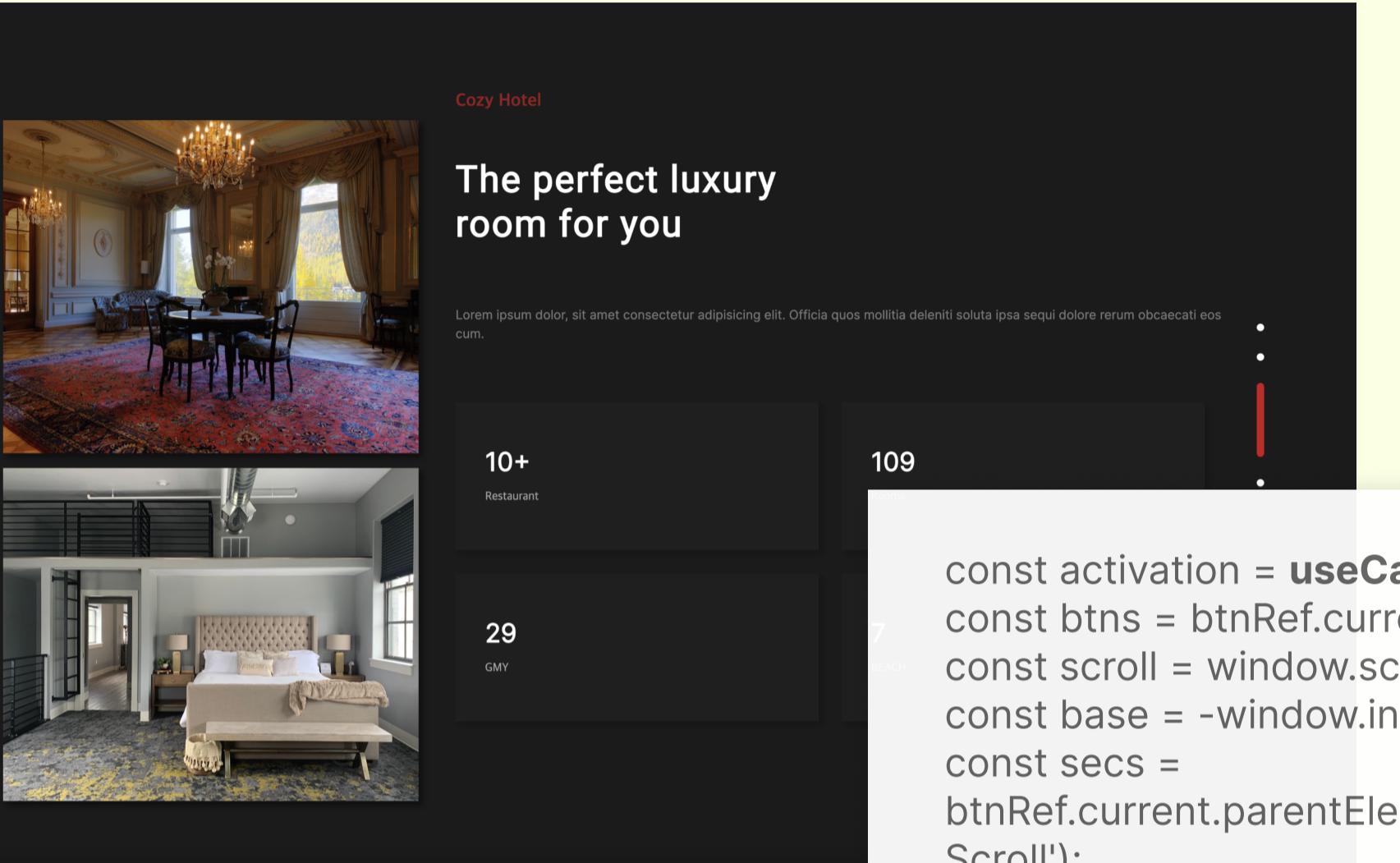
      btns[idx].classList.add('on');
      secs[idx].classList.add('on');
    }
  });
});
```

스크롤 값이 현재 섹션의 2/3이상 넘어가면 버튼과 섹션의 onclass 활성화를 걸어주는 기능

```
useEffect(() => {
  getPos();
  window.scrollTo({ top: 0, left: 0, behavior: 'smooth' });
  window.addEventListener('resize', getPos);
  window.addEventListener('scroll', activation);
  return () => {
    window.removeEventListener('resize', getPos);
    window.removeEventListener('scroll', activation);
  };
}, [getPos, activation]);
```

윈도우 리사이즈시 각 배열값을 다시 구하고 난뒤 클린업 함수를 사용하여서 기능 삭제

Rooms 페이지



asset/animate.js

```
onClick={() => {
  new Anim(window, {
    prop: 'scroll',
    value: pos.current[idx],
    duration: speed.current,
  });
}}
```

onClick시 animation plugin을 사용하여 이동

Rooms.js

```
const [scrolled, setScrolled] = useState(0);
const [pos, setPos] = useState([]);
return (
<>
<RoomFirstTab scrolled={scrolled} pos={pos[1]} />
<RoomSecondTab scrolled={scrolled} pos={pos[2]} />
<RoomThirdTab scrolled={scrolled} pos={pos[3]} />
<RoomBanner scrolled={scrolled} pos={pos[4]} />
<Btns setScrolled={setScrolled} setPos={setPos} />
</>
```

Btns의 state값을 부모 컴포넌트로 이동시켜 set값에 담은후 각 섹션 컴포넌트로 값을 전달

Room(section).js

```
export default memo(Room(section));
```

스크롤시 각 섹션의 재랜더링 방지하기위해 memo 처리

```
const activation = useCallback(() => {
  const bntRef = btnRef.current.children;
  const scroll = window.scrollY;
  const base = -window.innerHeight / 3;
  const secs =
    btnRef.current.parentElement.querySelectorAll('.my
  Scroll');
  setScrolled(scroll);

  pos.current.forEach((pos, idx) => {
    if (scroll >= pos + base) {
      for (const btn of bntRef) btn.classList.remove('on');
      for (const sec of secs) sec.classList.remove('on');

      bntRef[idx].classList.add('on');
      secs[idx].classList.add('on');
    }
  });
});
```

스크롤 값이 현재 섹션의 2/3이상 넘어가면 버튼과 섹션의 onclass 활성화를 걸어주는 기능

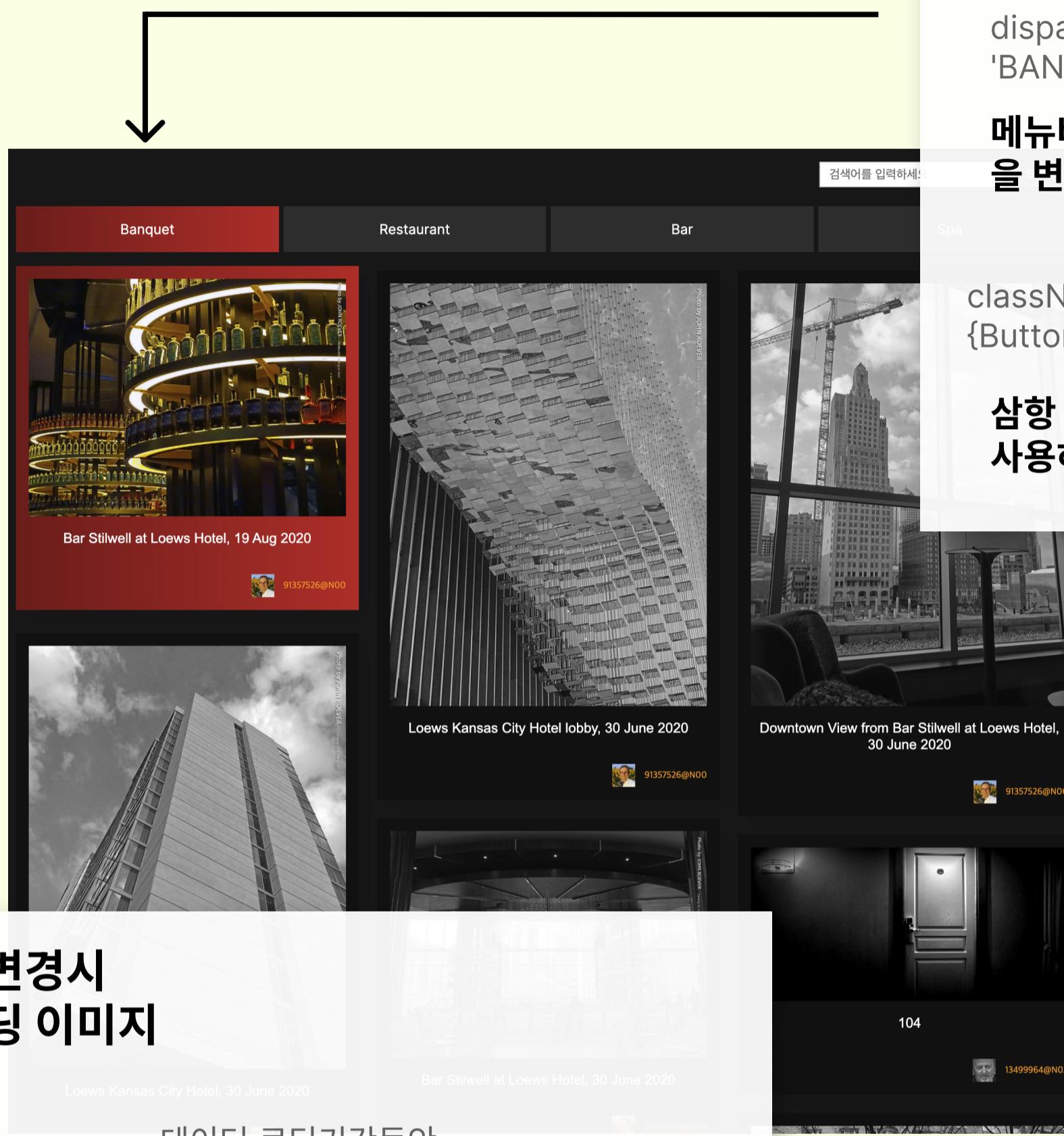
Facilities 페이지



Facilities.js

Flicker API를 사용하여
구현하였으며 데이터 관리는
유튜브 비디오와 같은 방법으로
redux-toolkit으로 관리함

이미지 클릭시
Modal창으로
이미지 띄우는 기능 구현



메뉴버튼 변경시 이미지 로딩 이미지

```
const Banquet = () => {
  frame.current.classList.remove('on');
  setLoading(true);
  dispatch(fetchFlickr({ type:
    'BANQUET' }));
};

setTimeout(() => {
  frame.current.classList.add('on')
  setLoading(false)
}, 500);
```



데이터 로딩기간동안
frame에 on클래스를 제거하여
안보이게 만든후 그동안
Loading 이미지를 나타낸다.

setTimeOut함수를 사용하여
0.5초후 frame에 on클래스를
붙히고 Loading 이미지를 사라
지게 한다

```
dispatch(fetchFlickr({ type:
  'BANQUET' }));
```

메뉴버튼 클릭마다 type값
을 변경하여 data값 변경

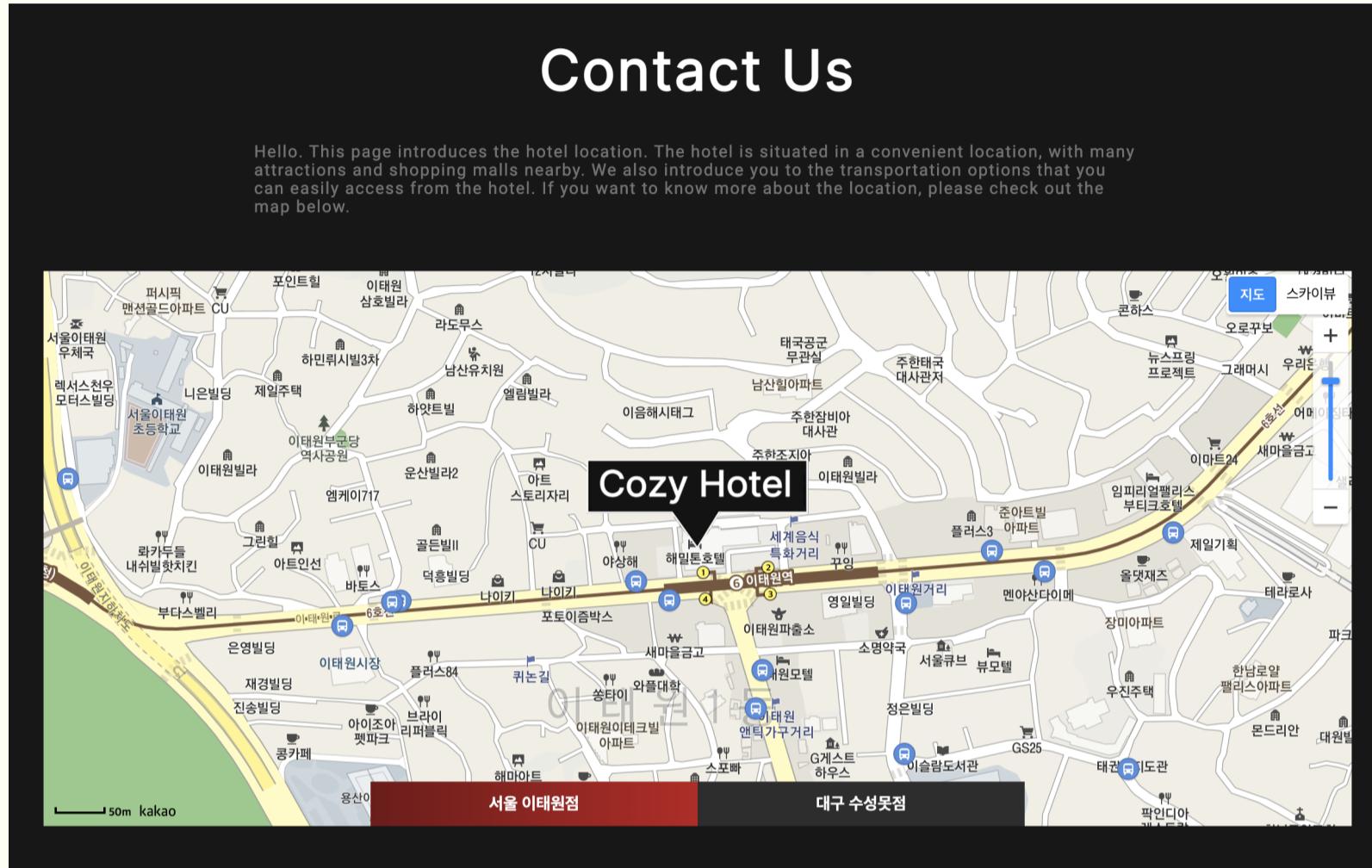
```
className= {Button === 'Banquet' ? 'on' : ''}
```

삼항 연산자로 onClass를
사용하여 현재 클릭된 효과

Contact 페이지

```
const option = useRef(null);
const info = useRef(null);
const container = useRef(null);
const { kakao } = window;
```

**지도를 가상돔으로 나타내는
객체들은 변하지 않는 정적인
정보이기 때문에 useRef 활용
kakao 객체는 비구조화 할당**



```
const setCenter = () => {
  map.setCenter(info.current[index].latlng);
};
```

지도 가운데를 유지시키는 함수

```
useEffect(()=> {
  window.addEventListener('resize', setCenter);

  return () => {
    window.removeEventListener('resize', setCenter);
  };
});
```

**윈도우 사이즈 리사이즈시 가운데
유지시키고 클린업 함수로 기능 삭제**

Contact 페이지

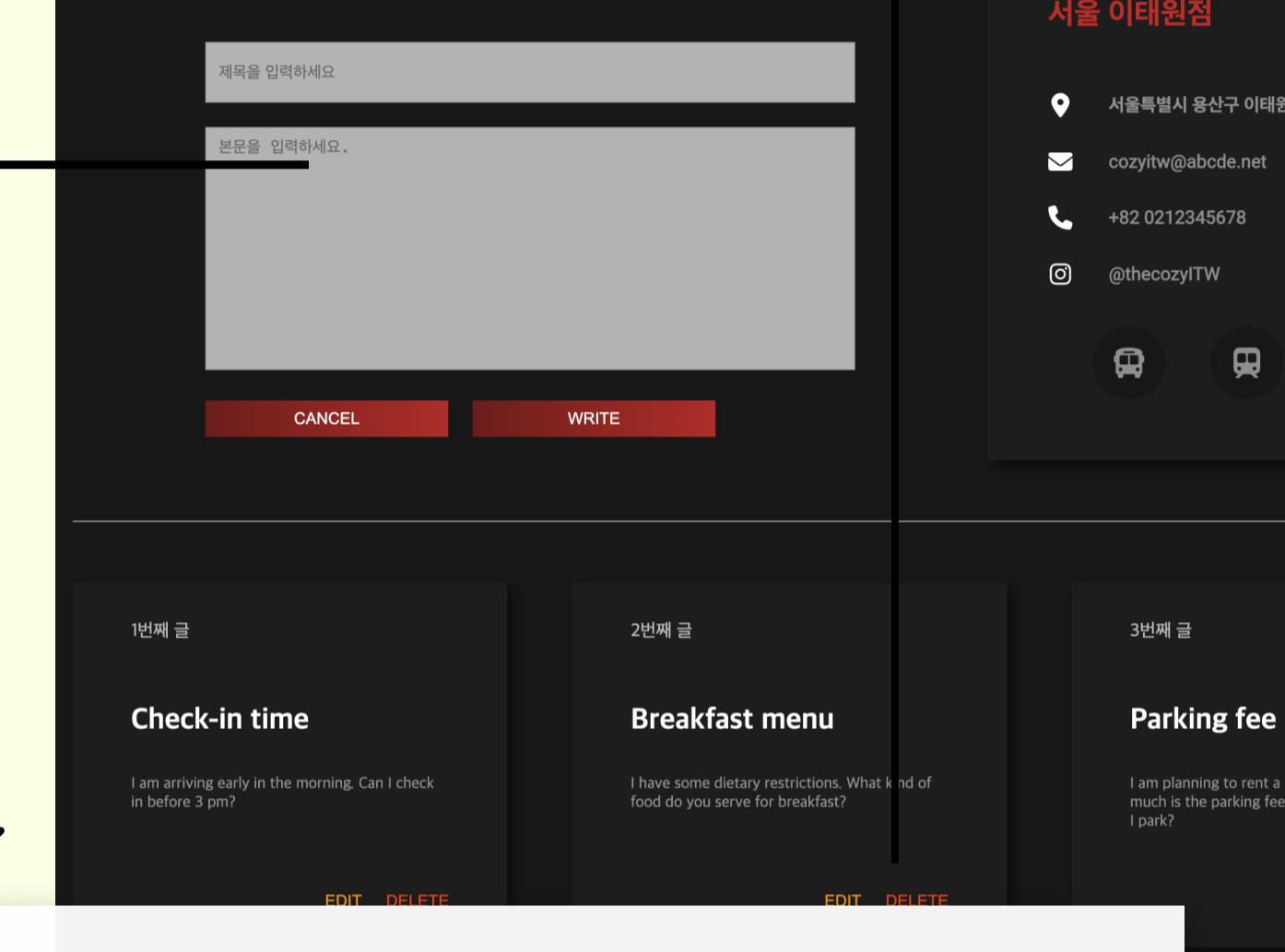
CRUD 기능

가상의 호텔기업페이지
Contact 페이지에 게시판

게시판을 통하여 글을 작성하고
작성한 글을 Localstorage에
담아서 출력하고,

작성된 글의
수정과 삭제기능을 구현하였다.

Post



Create

```
const createPost = () => {
  if (!input.current.value.trim() || !textarea.current.value.trim()) {
    resetForm();
    return alert('제목과 본문을 모두 입력하세요.');
  }
}
```

공백 방지 기능

```
setPosts([{ title: input.current.value, content:
  textarea.current.value }, ...Posts]);
resetForm(); 포스팅 이후 인풋 빙칸 초기화
};
```

Read

```
const getLocalData = () => {
  const dummies = [...(더미데이터)];
  const data = localStorage.getItem('post');
  if (data) return JSON.parse(data);
  else return dummies;
};
```

메인 페이지의
News.js 컴포넌트에서
데이터가 없을 때 더미데이터를
담아 놓은 것을 출력

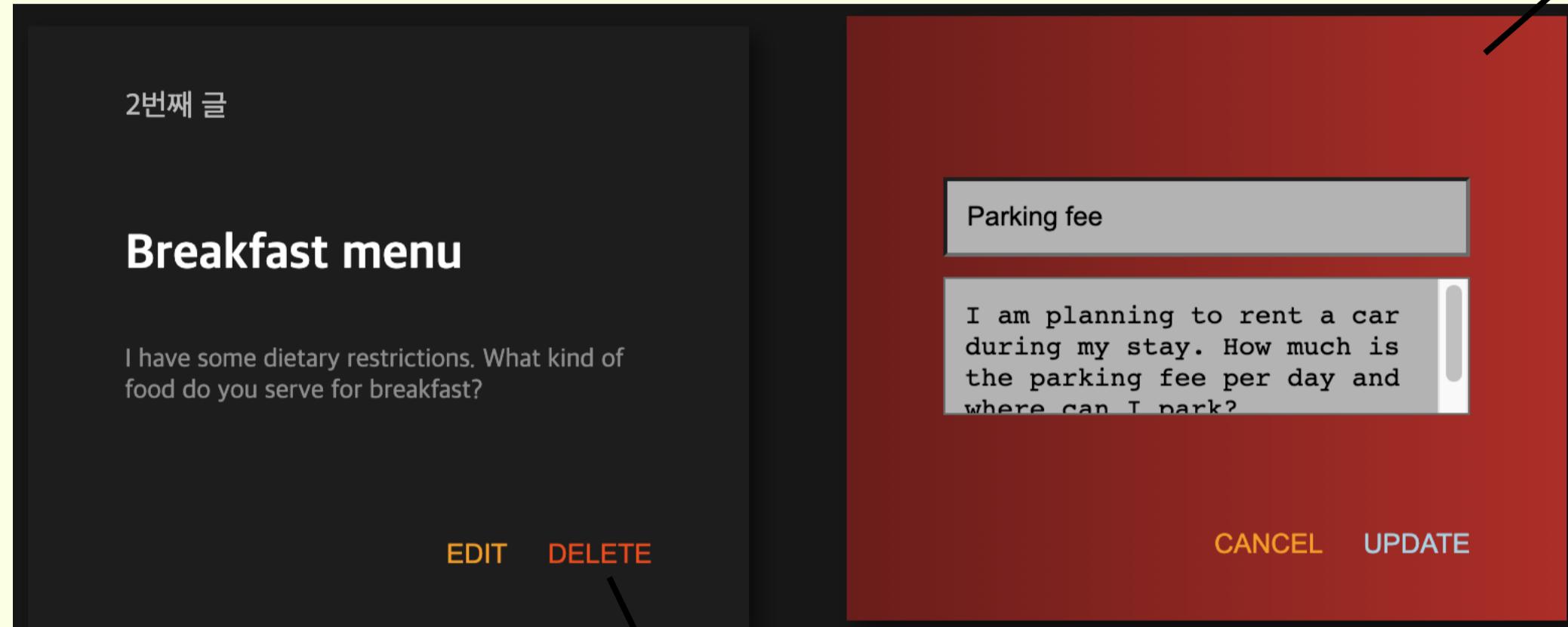
```
const getLocalData = () => {
  const data =
    localStorage.getItem('post');
  return JSON.parse(data);
};
```

Create 기능 후 localStorage의
새로운 Post가 담긴다면
더미데이터 이후로 출력

```
useEffect(() => {
  localStorage.setItem('post',
  JSON.stringify(Posts));
}, [Posts]);
```

useEffect로 Post
스테이트의 변경이 생길 때마다
localStorage에 저장

Contact 페이지



```
const deletePost = (delIndex) => {
  if (!window.confirm('해당 게시글을 삭제하겠습니까?')) return;
  setPosts(Posts.filter(_, postIndex) => postIndex !== delIndex));
};
```

삭제될 글에 index를 전달하여
filter로 기존 글index순번값과 같지않은 나머지글만 걸러냄

Update

```
const enableUpdate = (editIndex) => {
  if (!Allowed) return;
  setPosts(
    Posts.map((post, postIndex) => {
      if (postIndex === editIndex) post.enableUpdate = true;
      return post;
    })
  );
  setAllowed(false);
};
```

수정모드로 전환

기존의 post순번과 index로 전달받은
순번이 같은 post객체에
enableUpdate = true 값을
가해서 반환

복수의 글 수정모드 방지기능

```
const enableUpdate = () => {
  if (!Allowed) return;
  setAllowed(false);
};
```

수정모드에 진입하면 Allowed(기존
은 true)값을 flase로 바꾸어 다른글
의 수정모드 진입을 방지

삼항연산자를 활용한 수정모드 인터페이스 진입

```
<article>
{post.enableUpdate ? (
  <>수정모드 input태그 상태</>
) : (
  <>h2 , p 태그 상태</>
)}
```

Join 페이지

User ID
아이디를 입력하세요
아이디는 7글자 이상, 영어, 숫자를 포함하여 입력하세요

Password
비밀번호를 입력하세요.
비밀번호는 10글자 이상, 영문, 숫자, 특수문자를 모두 포함하세요

Re Password
비밀번호를 재 입력하세요
두개의 비밀번호를 동일하게 입력하세요

E-mail
이메일주소를 입력하세요
이메일은 8글자 이상, @를 포함하세요

Country
선택하세요
국적을 선택하세요

Address
시/군/구를 적어주세요
아파트주소를 적어주세요 (선택)
주소를 정확히 기입하세요

ZIP-CODE
우편번호를 적어주세요
우편번호 5글자를 입력해주세요

Gender
Male ● Female ●
성별을 선택하세요

Comments
남기는 말을 입력하세요
남기는 말은 20글자 이상 입력하세요.

cancel submit

Contact Us

Lorem ipsum dolor sit amet consectetur adipisicing elit. Architecto quae doloribus laudantium cum, nostrum iste veniam minus rerum quasi sed.

E-mail : abcdefg@gmail.com
Phone : (+82)10-1234-5678

검증완료

```
useEffect(() => {
  const len = Object.keys(err).length;
  if (len === 0 && submit.current) {
    alert('진심으로 환영합니다!');

    history.push('/');
  }
}, [err, history]);
```

에러값이 없다면
alert창이 뜨고
화면이 이동하며
회원가입 완료

회원가입 검증 기능

```
if (value.userid.length < 7 || !eng.test(value.userid) || !num.test(value.userid)) {
  errs.userid = '아이디를 7글자 이상, 영어, 숫자를 포함하여 입력하세요';
}

if (
  value.pwd1.length < 10 ||
  !eng.test(value.pwd1) ||
  !num.test(value.pwd1) ||
  !spc.test(value.pwd1)
) {
  errs.pwd1 = '비밀번호는 10글자 이상, 영문, 숫자, 특수문자를 모두 포함하세요';
}
if (value.pwd1 !== value.pwd2 || !value.pwd2) {
  errs.pwd2 = '두개의 비밀번호를 동일하게 입력하세요';
}

if (value.email.length < 8 || !/@/.test(value.email)) {
  errs.email = '이메일은 8글자 이상, @를 포함하세요';
}

if (!value.gender) {
  errs.gender = '성별을 선택하세요.';
}
if (value.country === "") {
  errs.country = '국적을 선택하세요.';
}
if (value.adr.length < 10) {
  errs.adr = '주소를 정확히 기입하세요';
}
if (value.zip.length !== 5) {
  errs.zip = '우편번호 5글자를 입력해주세요';
}
if (value.comments.length < 20) {
  errs.comments = '남기는 말은 20글자 이상 입력하세요.';
}
```

감사합니다