



에어비앤비 웹사이트 제작

유혜경



CATEGORI

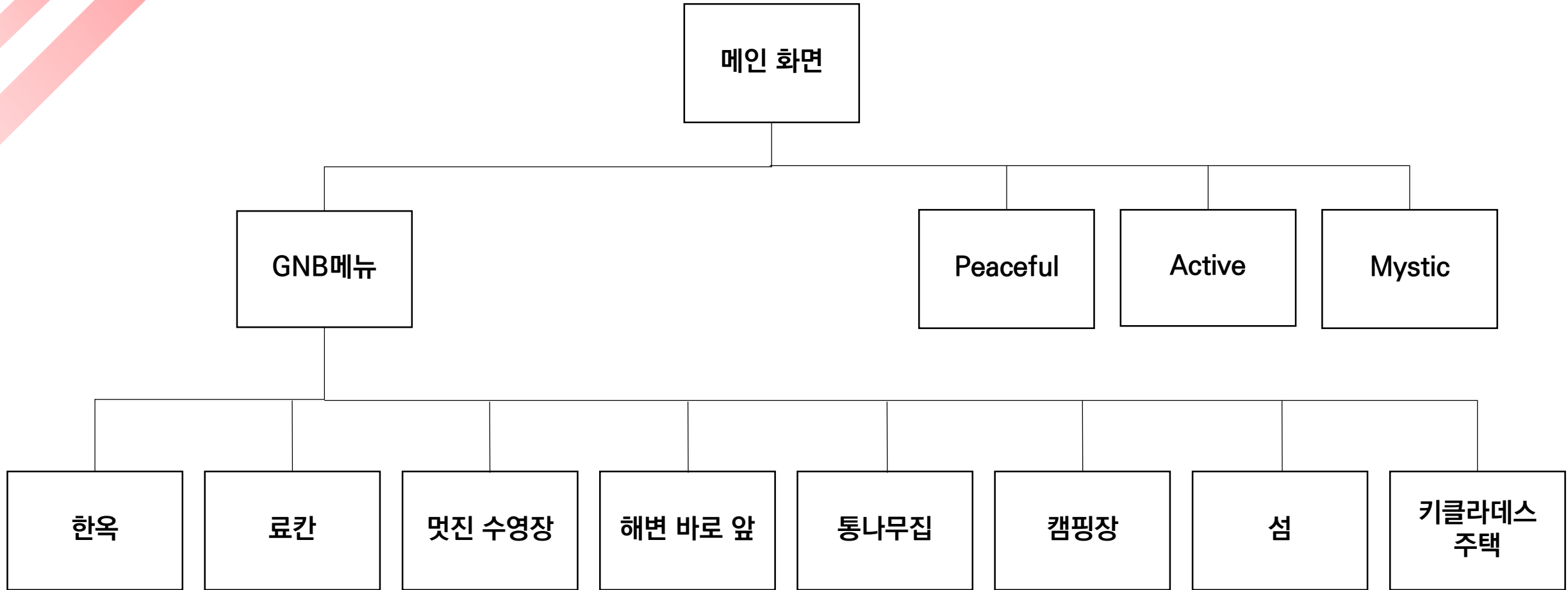
01 사이트 설계

02 주요 구현 항목

03 모바일 설계 / 개선사항

04 추가 구현 항목

01 사이트 설계



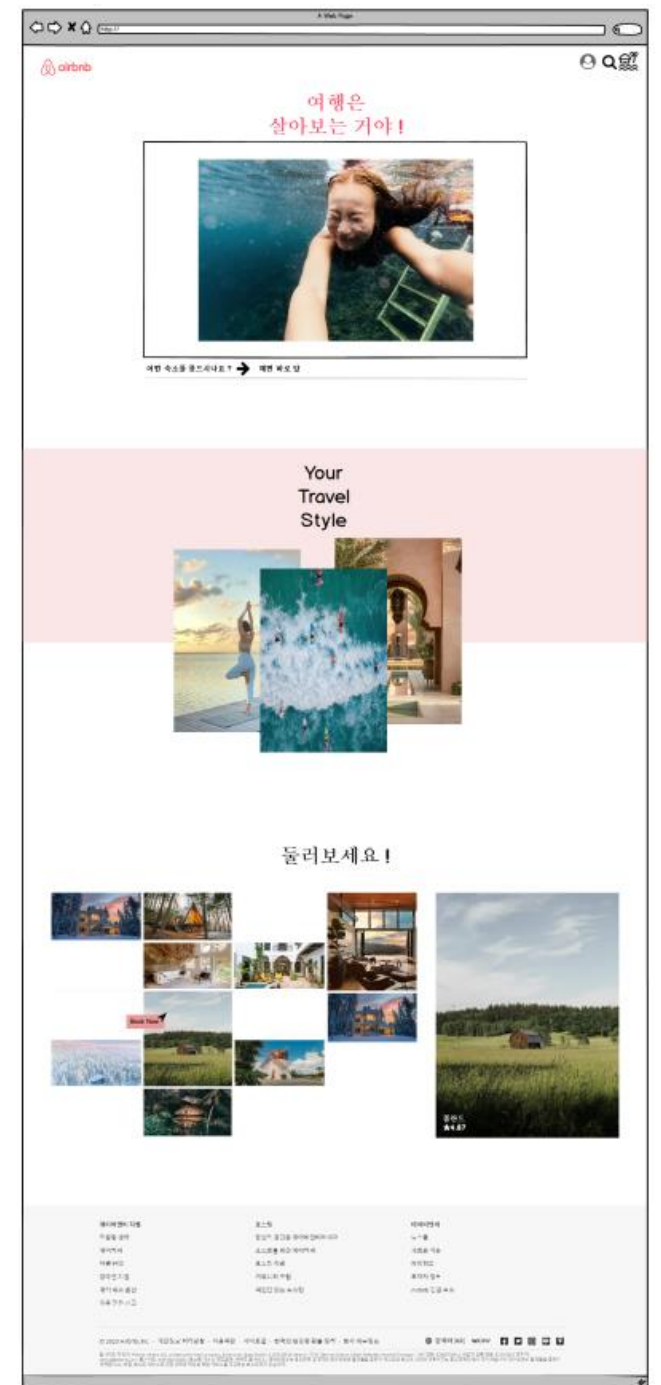
메인 화면 - 카테고리를 담은 GNB메뉴 / 각 스타일에 맞는 카테고리로 랜덤이동하는 버튼 3개

-> 메인페이지에서 이동시 Link to로 연결하여 라우터에 연결된 페이지 이동

GNB메뉴 - 총 8개의 카테고리로 구성

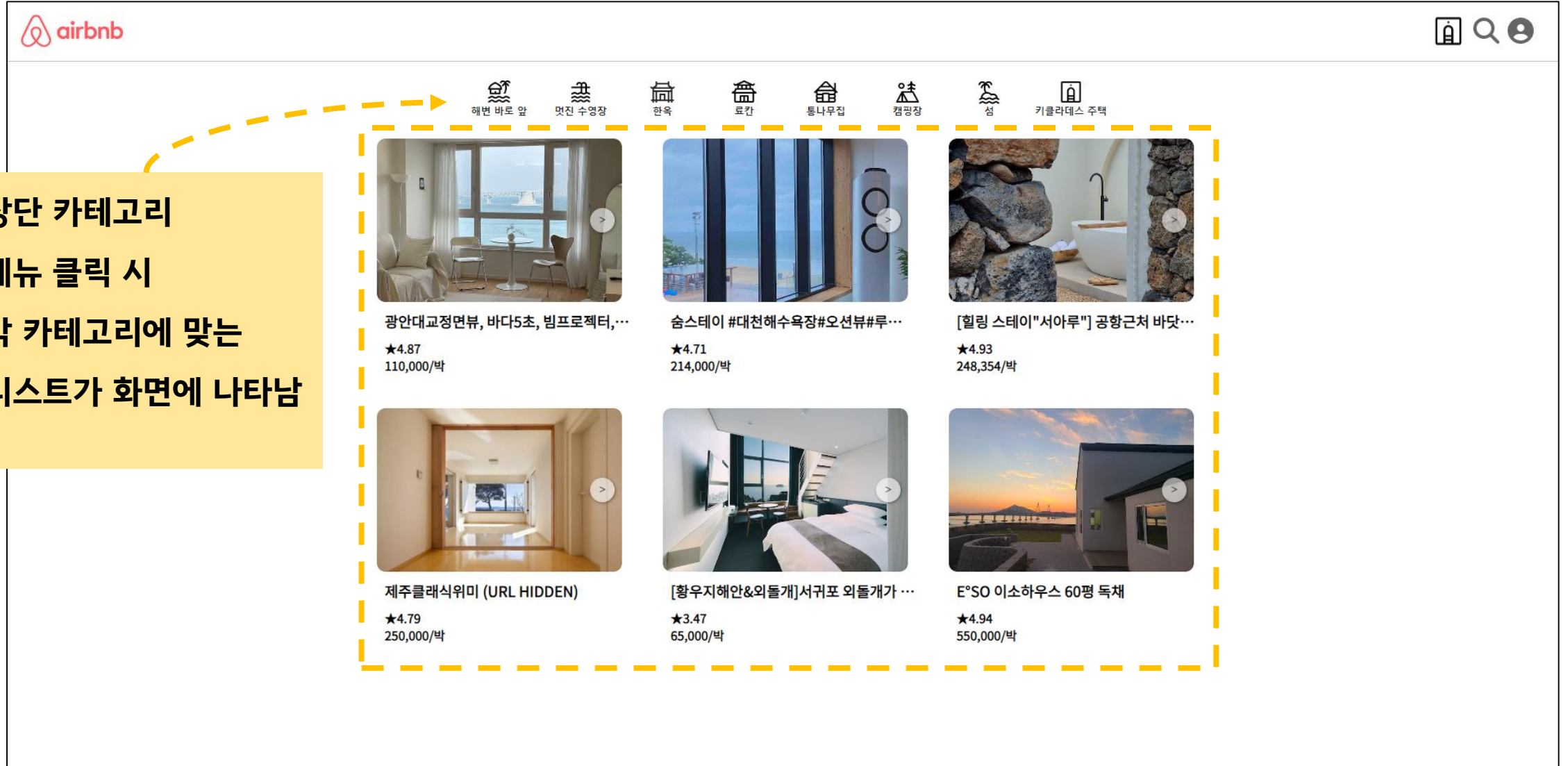
-> 카테고리 이동시 페이지 변경 없이 List라는 하나의 컴포넌트로 데이터만 새로 뿌려줄 수 있도록 구현

기존 사이트는 메인 화면 없이 바로 카테고리 페이지가 나타남
메인 화면을 따로 구성하고 카테고리 메뉴와 연결



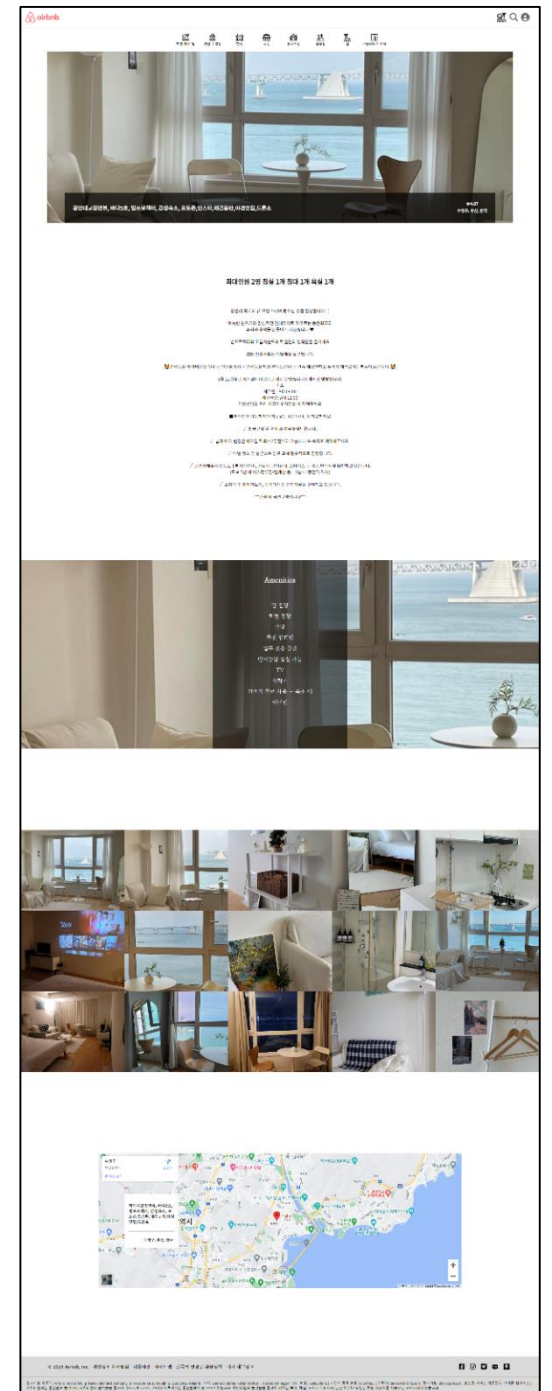
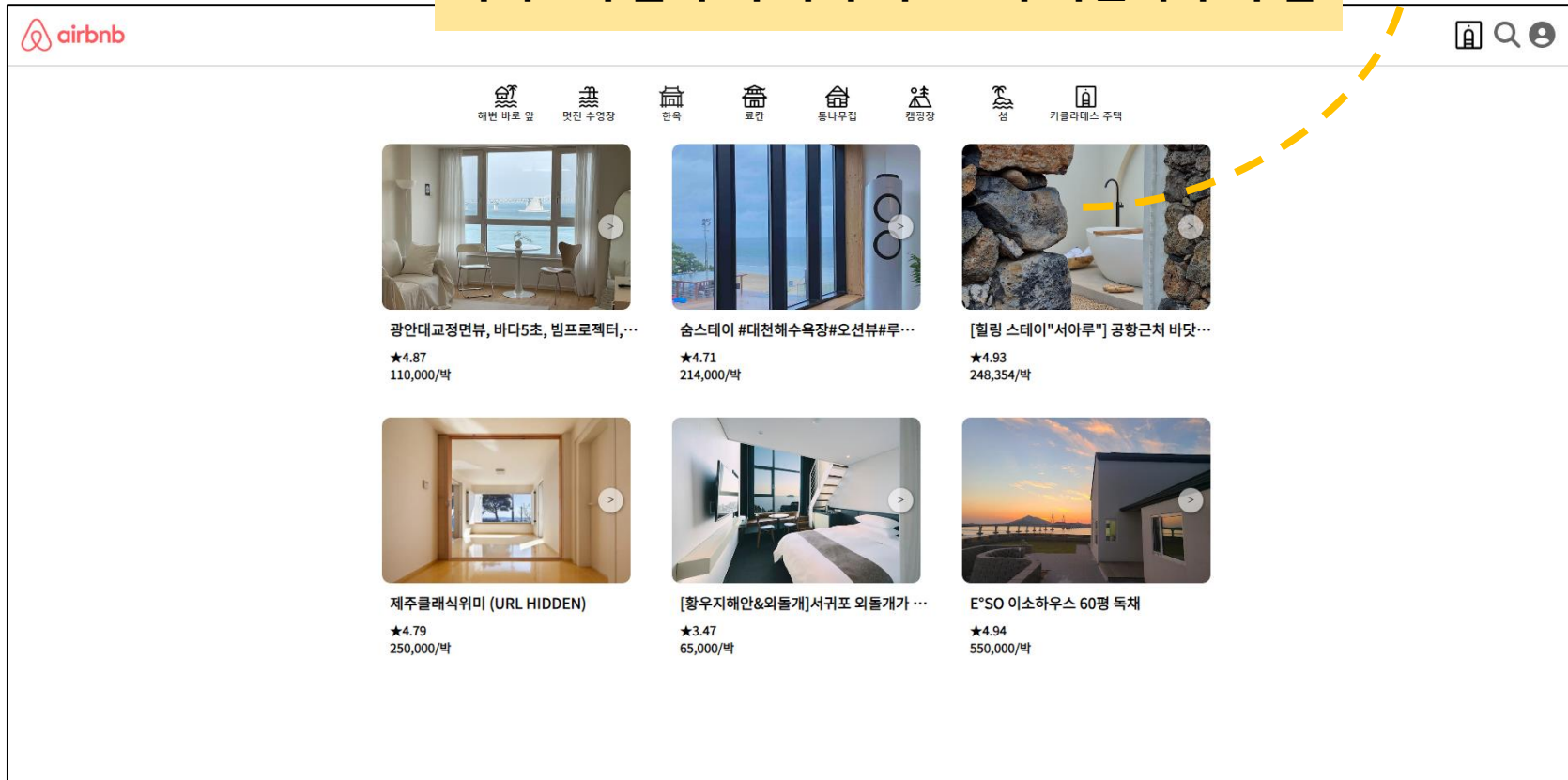
01 사이트 설계

상단 카테고리
메뉴 클릭 시
각 카테고리에 맞는
리스트가 화면에 나타남



01 사이트 설계

리스트의 각 이미지 클릭 시
세부 페이지로 이동, 세부 페이지 안에서도
카테고리 메뉴는 계속 렌더링 되어
카테고리 클릭 시 다시 리스트가 화면에 뿌려 짐



02 주요 구현 항목

```
export function Layout() {  
  const [menuCat, setMenuCat] = useState("해변 바로 앞");  
  
  const chgMenuCat = txt => {  
    setMenuCat(txt);  
  }  
  
  const [bNum, setBNum] = useState(100);  
  
  return(  
    <bnbCon.Provider value={{menuCat, chgMenuCat, bNum, setBNum}}>  
      <TopArea/>  
      <MainArea/>  
      <FooterArea/>  
    </bnbCon.Provider>  
  )  
} /////////////// Layout ///////////////
```

```
export function List(props) {  
  // props.cat은 전달된 카테고리 정보(데이터 선별용)  
  const selData = catData.filter((v) => {  
    if (v.category == decodeURIComponent(props.cat)) return true;  
  });  
  // console.log(selData);  
}
```

```
export function CatMenu(){  
  const myCon = useContext(bnbCon);  
  const goNav = useNavigate();  
  
  // console.log(menu);  
  return(  
    menu.map((v, i) => (  
      <li className="cat-menu" key={i} onClick={(e)={myCon.chgMenuCat(v.txt);  
        e.stopPropagation();goNav('/list')}}>  
        <img src={v.img} alt={v.txt}/><br/>{v.txt}  
      </li>  
    ))  
  )  
} /////////////// showCat ///////////////
```

[카테고리 메뉴]

카테고리 데이터와 리스트 데이터를 usecontext 로 가져옴

useState로 리스트의 상태 변수를 만들고

카테고리 메뉴에 연결된 txt 데이터와 category를 일치 시켜 리스트가 화면에 뿌려지도록 함

02 주요 구현 항목

```
return (  
  <section className="cat-list">  
    {selData.map((v, i) => (  
      <Link to="/detail" key={i}  
        state={{  
          name: v.name,  
          score: v.score,  
          idx: v.idx,  
          cat: v.category,  
          seq: i,  
        }}>  
    )}
```

```
export function Detail() {  
  const loc = useLocation();  
  
  const idx = loc.state.idx;  
  const name = loc.state.name;  
  const score = loc.state.score;  
  const cat = loc.state.cat;  
  
  let seq = loc.state.seq;
```

```
return (  
  <>  
    /* 1. 메인사진 파트 */  
    <div className="main-img">  
      /* 메인이미지 : 첫번째 이미지*/  
      <img src={`./images/Category/" + cat + "/" + (seq + 1) + "/1.jpg`} />  
      <div className="namebx">  
        /* 숙소이름 : catData - name*/  
        <h1>{name}</h1>  
        /* 숙소평점 : catData - score*/  
        <ul>  
          <h3>★{score}</h3>  
          /* 숙소위치 : subData - gps*/  
          <h3>{selData.gps}</h3>  
        </ul>  
      </div>  
    </div>  
  </div>
```

[상세 페이지 구성]

리스트 컴포넌트에서 카테고리 및 일치시켜 뿌려진 데이터가 상세 페이지에서 재사용 되므로
uselocation으로 사용할 데이터를 state에 담아 가져온 후 사용함

03 모바일 설계



GNB메뉴를 세로형 으로 변경

리스트는 화면 크기에 맞춰 그리드 설정

03 모바일 설계 - 완료



메인 화면

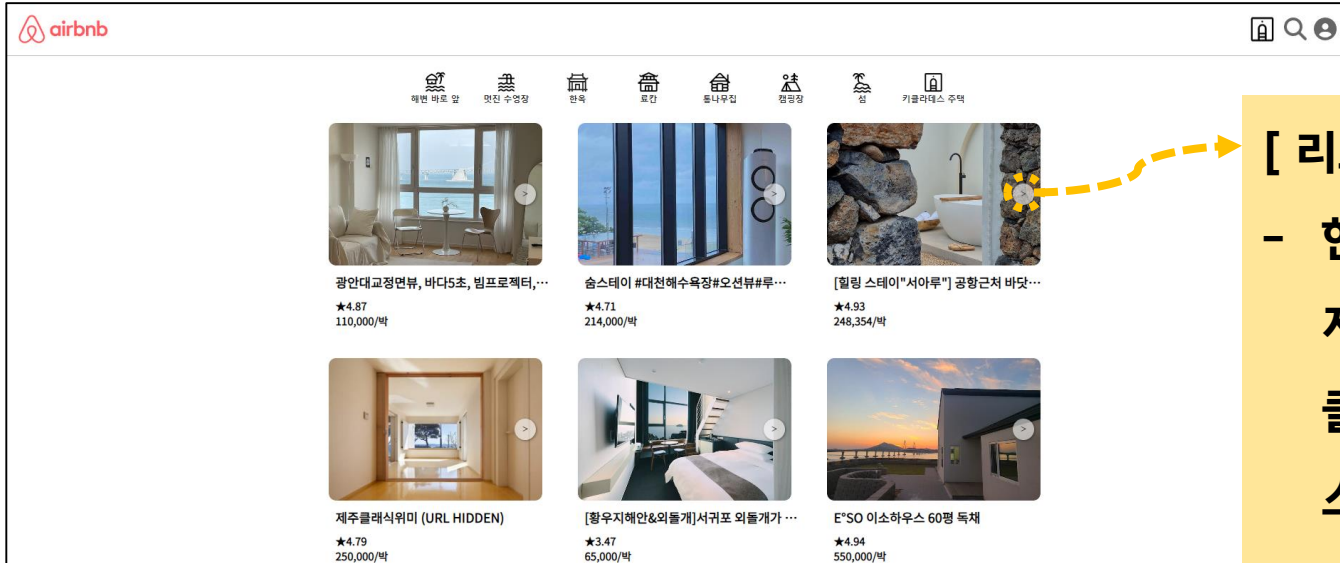


GNB 메뉴

각 페이지 별
미디어 쿼리가 적용됨

GNB메뉴는 모바일일
경우 오른쪽에서 왼쪽으로
등장하도록 변경하였음

개선 사항



[리스트 이미지 스와이프 기능]

- 한 카테고리에서 버튼을 클릭해서 이미지를 스와이프 하면 다른 카테고리를 클릭시 같은 위치의 이미지가 똑같이 스와이프 되어있는 문제 발생
- 카테고리가 변경될 때 마다 스와이프가 초기화 될 수 있도록 변경

Your Travel Style



Peaceful



Active

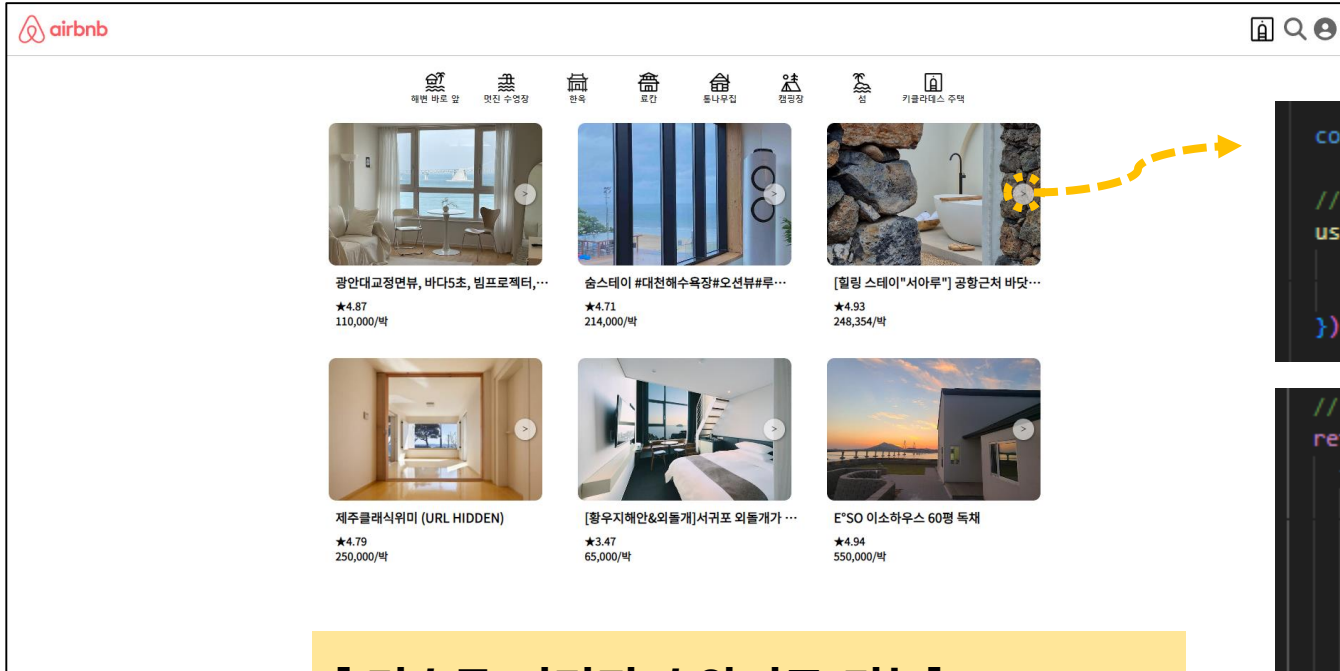


Mystic

[랜덤 카테고리로 이동하는 기능]

- 중복된 난수의 발생으로 여러 번 같은 카테고리 리스트가 출력됨
- While문을 통한 중복 제거로 매번 다른 페이지가 나오게 구현

개선 사항 - 완료



```
const myRef = useRef(null);

// 렌더링 후 적용(스와이퍼 초기화)
useEffect(()=>{
  // console.log(myRef);
  myRef.current.swiper.slideTo(0,0);
})
```

```
// 리턴코드 //////////////////////////////////
return (
  <>
    <Swiper
      ref={myRef}
      /* ref 속성에 useRef 할당변수를 넣어서 외부에 연결함 */
      slidesPerView={1}
      spaceBetween={0}
      navigation={true}
      /* 사용할 모듈을 여기에 적용시킨다 */
      modules={[Navigation]}
      className="mySwiper"
    >
      {makeList()}
    </Swiper>
  </>
);
//////////////////////////////// SwiperApp 컴포넌트 //////////////////////////////////
```

[리스트 이미지 스와이프 기능]

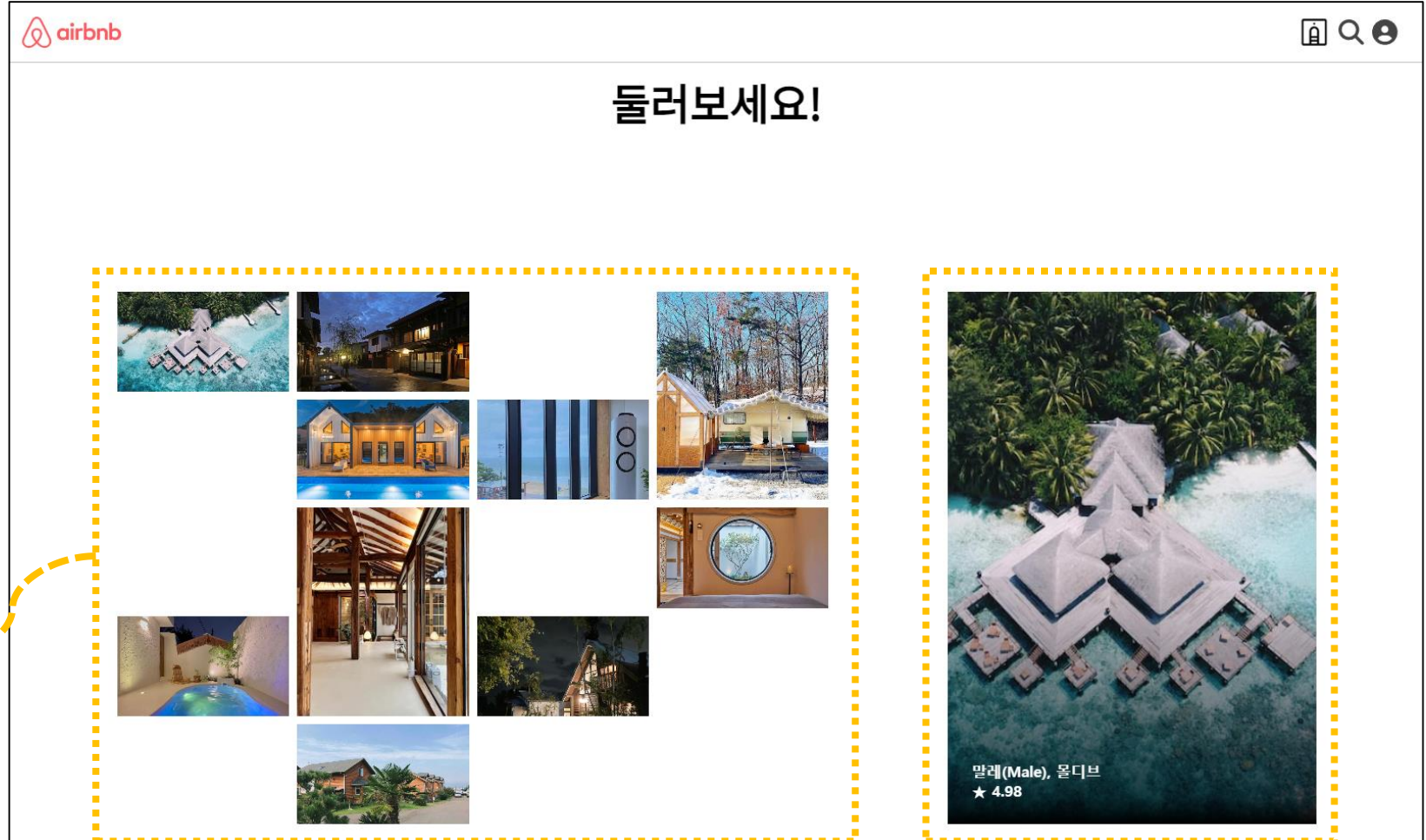
리스트 페이지가 렌더링 될 때 스와이퍼가 초기화 될 수 있도록 ref값을 이용하여 useEffect구역에서 적용하였음

04 추가 구현 항목

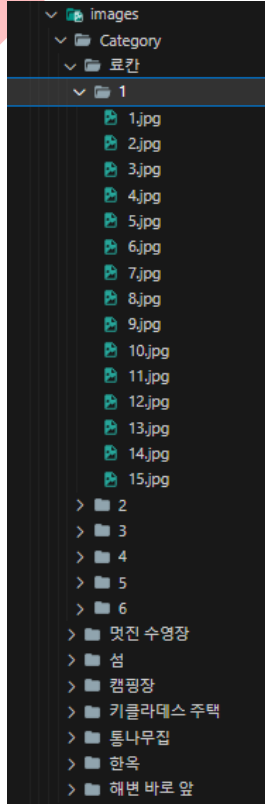
[메인 페이지 세번째 콘텐츠 파트]

페이지가 랜더링 될 때 마다 랜덤으로 숙소 이미지와 숙소 정보가 추출되어 10개씩 뿌려짐

각 이미지는 마우스 오버시 오른쪽에 큰 이미지와 숙소 정보가 나타남



04 추가 구현 항목



```
41 const filterVal = (val) => {
42   let res = false;
43   // console.log("필터에서 val:", val);
44   // console.log("필터에서 arr:", arr);
45
46   arr.forEach((v) => {
47     if (v === val) res = true;
48   });
49
50   // console.log("검사결과:", res);
51   return res;
52 };
53
54 /// for문으로 10개를 만들어 arr배열에 넣기 ////
55 const makeArr = () => {
56   for (let x = 0; x < 10; x++) {
57     let nowRdm = retRdm();
58     // console.log(Array.isArray(arr));
59     while (filterVal(nowRdm)) {
60       nowRdm = retRdm();
61       // console.log("다시만들어!");
62     } // while
63
64     // 통과후 반영하기
65     arr.push(nowRdm);
66   } // for
67 }; // makeArr
```

```
104 const arrNum = [2, 4, 7, 8, 13, 14, 16, 17];
105 const chkNum = (x) => {
106   // console.log('입력:', x);
107   let retVal = true;
108   arrNum.forEach((v) => {
109     if (v === x) retVal = false;
110     // console.log('틀마!', v);
111   });
112
113   return retVal;
114 }; // chkNum
115
116 const makeList = () => {
117   let temp = [];
118
119   let setNum = -1;
120
121   for (let i = 0; i < 18; i++) {
122     // console.log(chkNum(i));
123     temp[i] = (
124       <li key={i}>
125         {chkNum(i) ? (
126           <img src={arr[++setNum]} alt="image" onMouseEnter={setEvt} />
127         ) : (
128           ""
129         )}
130       </li>
131     );
132   }
133
134   return temp;
135 }; // makeList
```

그리드 형식에 맞춘 구조로,
이미지가 없는 빈칸 부분에도
를 생성해야 함

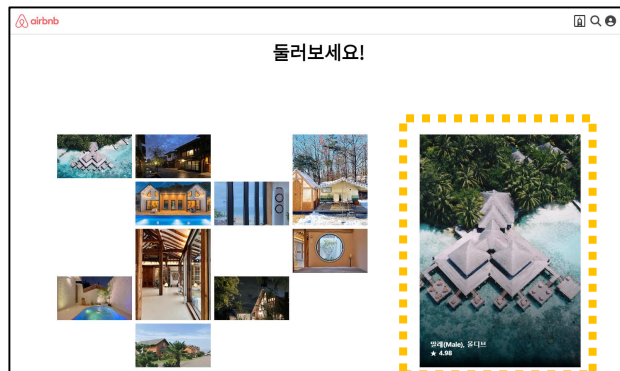
-> for문으로 총 18개 li를
생성하되, 삼항연산자를 이용해
이미지가 없는 빈 li를 생성

이미지 경로 데이터를 따로 만들지 않고 파일 경로를 따라서
src를 빈 배열 arr에 중복없이 담아 놓음
총 카테고리 8개 / 폴더 번호 6개를 각각 랜덤으로 추출
src : “./image/Category/카테고리이름/폴더번호/1.jpg”

```
25 const retRdm = () => {
26   // 0~7 난수
27   let rdm1 = Math.floor(Math.random() * 8);
28   // 1~6 난수
29   let rdm2 = Math.ceil(Math.random() * 6);
30
31   // 새로운 조합 src
32   let newSrc =
33     "./images/Category/" +
34     catTxt[rdm1].replace(/s/gi, "%20") +
35     "/" +
36     rdm2 +
37     "/1.jpg";
38   return newSrc;
39 };
40
```

04 추가 구현 항목

```
73 const setEvt = (e) => {  
74   const tgEle = $(e.currentTarget);  
75   const csrc = tgEle.attr("src");  
76   console.log("현재src:", csrc);  
77  
78   const temp = csrc.split("/");  
79   console.log(temp[3], temp[4]);  
80  
81   const fdata = catData.find((v) => {  
82     if (  
83       v.category === temp[3].replace(/%20/gi, " ") &&  
84       v.icat === Number(temp[4])  
85     )  
86       return true;  
87     });  
88  
89   console.log("선택Data:", fdata);  
90  
91   const loca = subData.find((v) => {  
92     if (v.idx === fdata.idx) return true;  
93   });  
94  
95   $(".bigimg")  
96     .css("background-image", `url(${csrc})`)  
97     .find("h3")  
98     .first()  
99     .text(loca.gps)  
100    .next()  
101    .text("★ " + fdata.score);  
102  }; // setEvt //
```



```
export const catData = [  
  {  
    "idx": '1',  
    "name": '광안대교정면뷰, 바다5초, 빔프로젝터, 감성숙소, 포토존,인',  
    "category": '해변 바로 앞',  
    "score": '4.87',  
    "price": '110,000',  
    "scat": ['해변바로앞'],  
    "icat": 1  
  },  
  {  
    "idx": '2',  
    "name": '숨스데이 #대천해수욕장#오션뷰#루프탑바베큐#프라이',  
    "category": '해변 바로 앞',  
    "score": '4.71',  
    "price": '214,000',  
    "scat": ['바베큐장,해변바로앞'],  
    "icat": 2  
  },  
  {  
    "idx": '3',  
    "name": '[힐링 스테이"서마루"] 공항근처 바닷마을, 독채 - 야외옥조',  
    "category": '해변 바로 앞',  
    "score": '4.93',  
    "price": '248,354',  
    "scat": ['마당', '옥조,해변바로앞'],  
    "icat": 3  
  },  
  // ...  
];
```

```
1 export const subData = [  
2   {  
3     idx: "1",  
4     gps: "수영구, 부산, 한국",  
5     room: "최대인원 2명 침실 1개 침대 1개 욕실 1개",  
6     detail: "광안리 최고의 뷰 모던 스테이에 오신 것을 환영합니다 : )  
7  
8     마늑한 분위기와 감성적인 인테리어로 머무르는 동안최고의  
9     휴식과 추억을 만들어 드리겠습니다 ♥  
10  
11     빔프로젝터와 고급사운드바로 들만의 영화관을 즐기세요  
12  
13     모든 침구세트는 매일매일 살균됩니다.  
14  
15     🐾반려동물 추가비용있습니다. 인원설정시 꼭 반려동물체크 부탁드립니다! 간혹  
16  
17     5월 21일부로 시스템에어컨으로 새로 달았습니다!! 에어컨 땡땡합니다!  
18     숙소  
19     체크인 : PM 04:00  
20     체크아웃: AM 12:00
```

2 [score 데이터 가져오기]

숙소 이미지와 정보를 매칭하기 위해
src에 담긴 카테고리(category) 및
폴더번호(icat) 정보를 catData와
비교하여 fdata에 담음

3 [gps 데이터 가져오기]

catData와 subData는 idx가 1~48로 연결
된 데이터 이므로 fdata에 담긴 idx와
subData의 idx를 비교하여 현재 숙소
이미지에 맞는 정보를 찾아서 loca에 담음

1

이미지의 src를 가져오기 위해
현재 mouse enter가 되어있는
이미지의 src 속성을 담아서 가져옴



감사합니다.