



**에어비앤비 웹사이트 제작**

**유혜경**



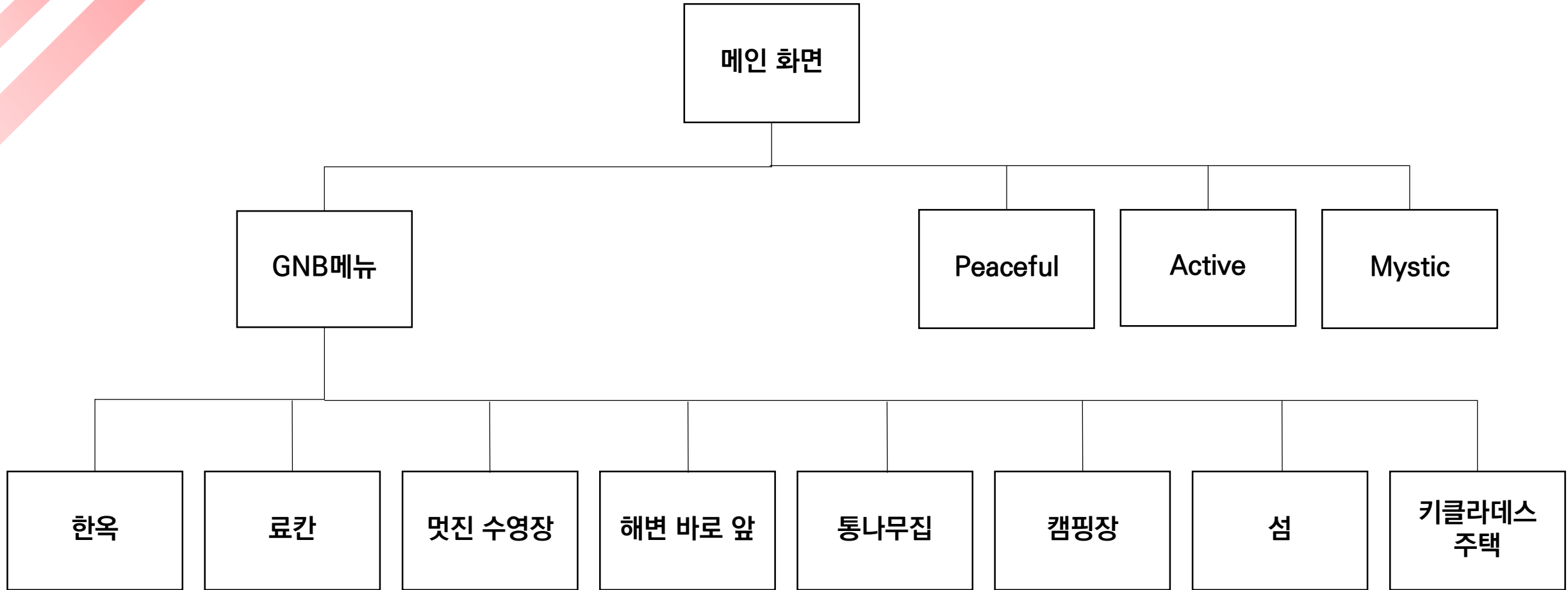
# CATEGORI

**01 사이트 설계**

**02 주요 구현 항목**

**03 모바일 설계**

# 01 사이트 설계



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

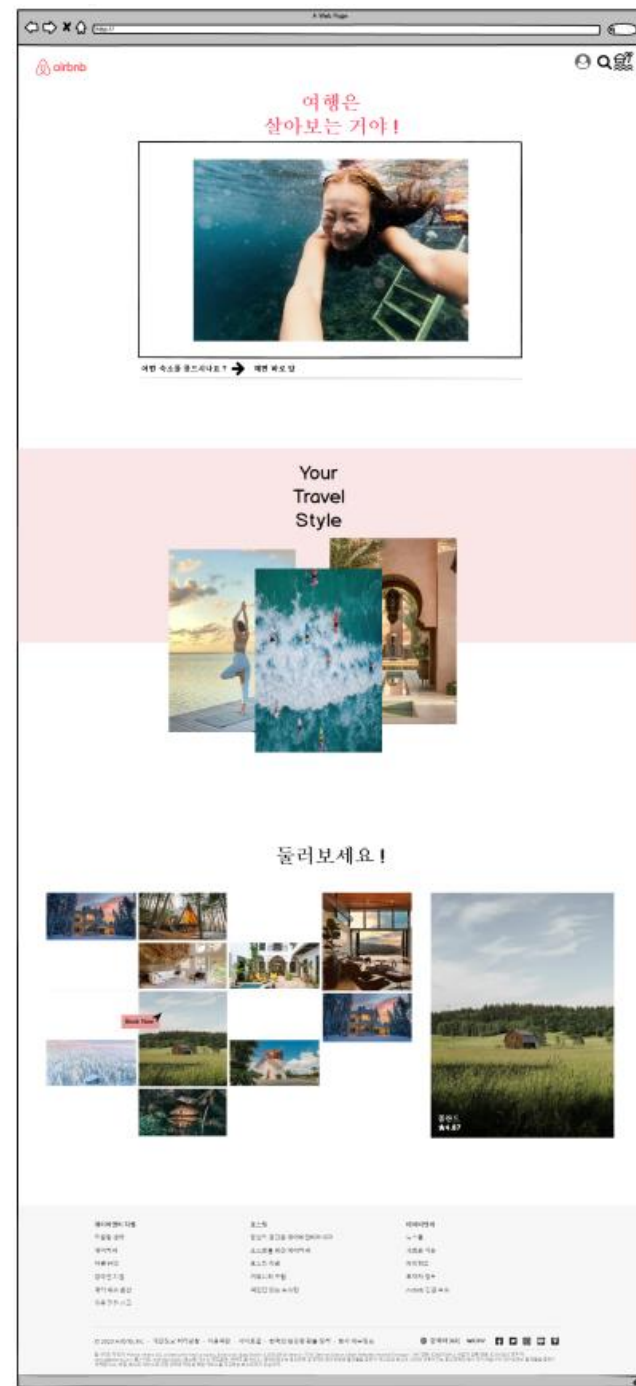
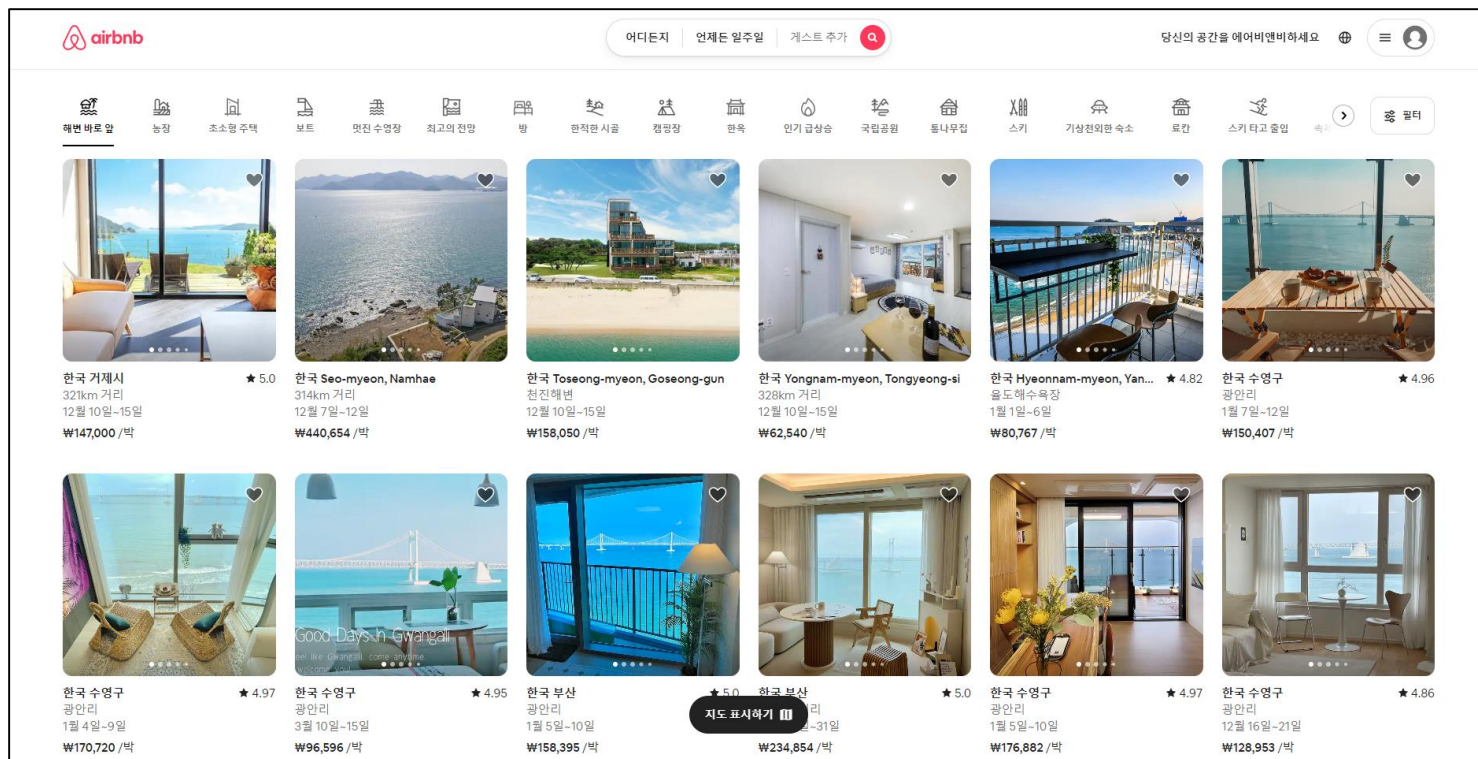
**-> 메인페이지에서 이동시 Link to로 연결하여 라우터에 연결된 페이지 이동**

**GNB메뉴 - 총 8개의 카테고리로 구성**

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

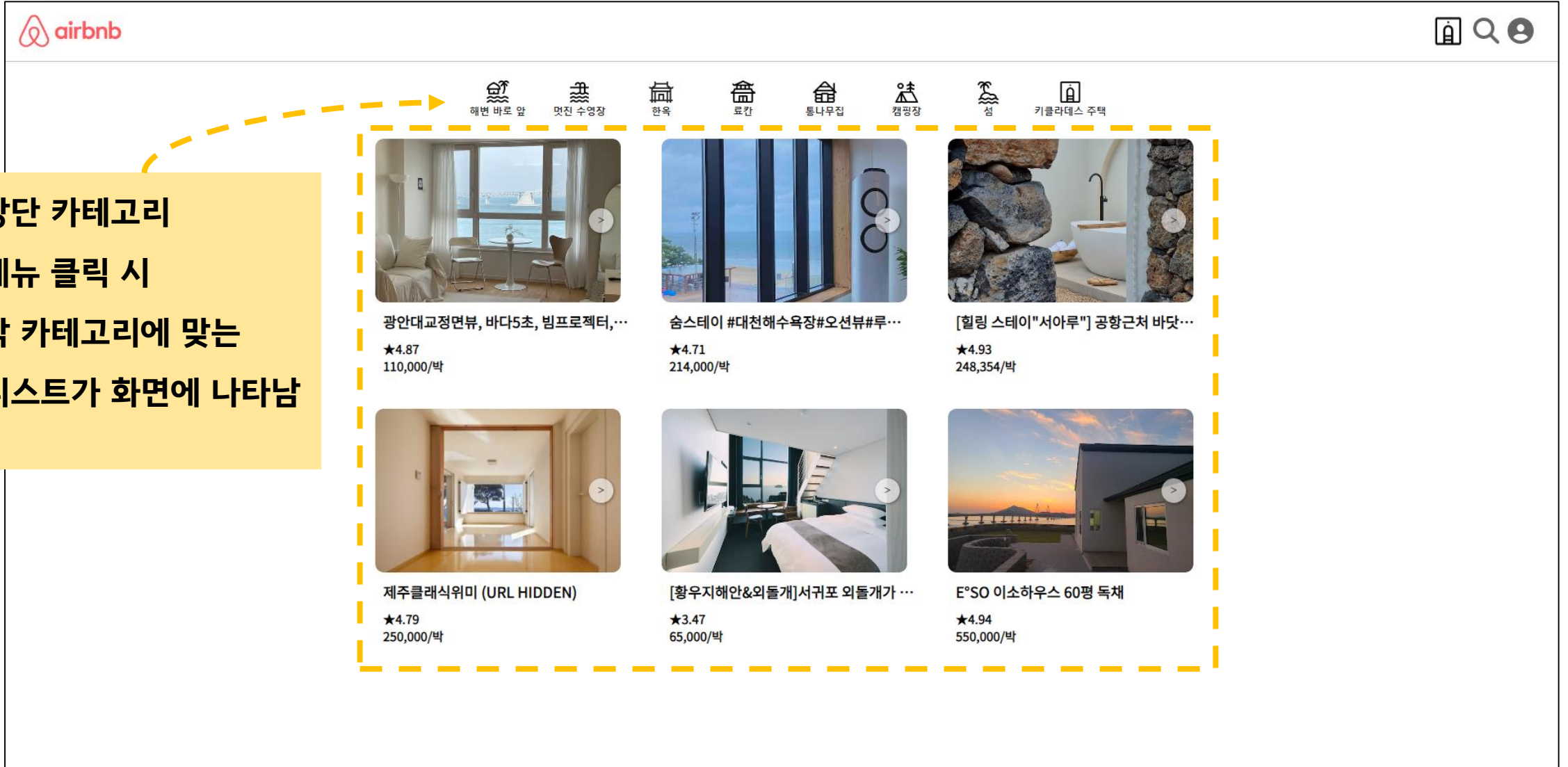
# 01 사이트 설계

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



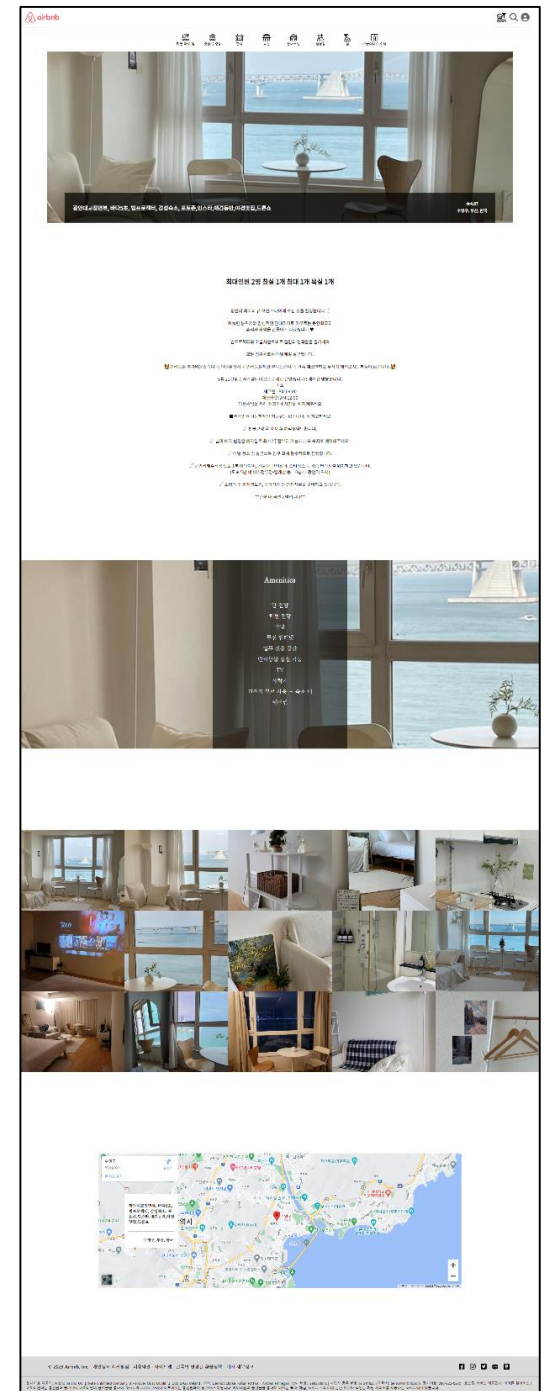
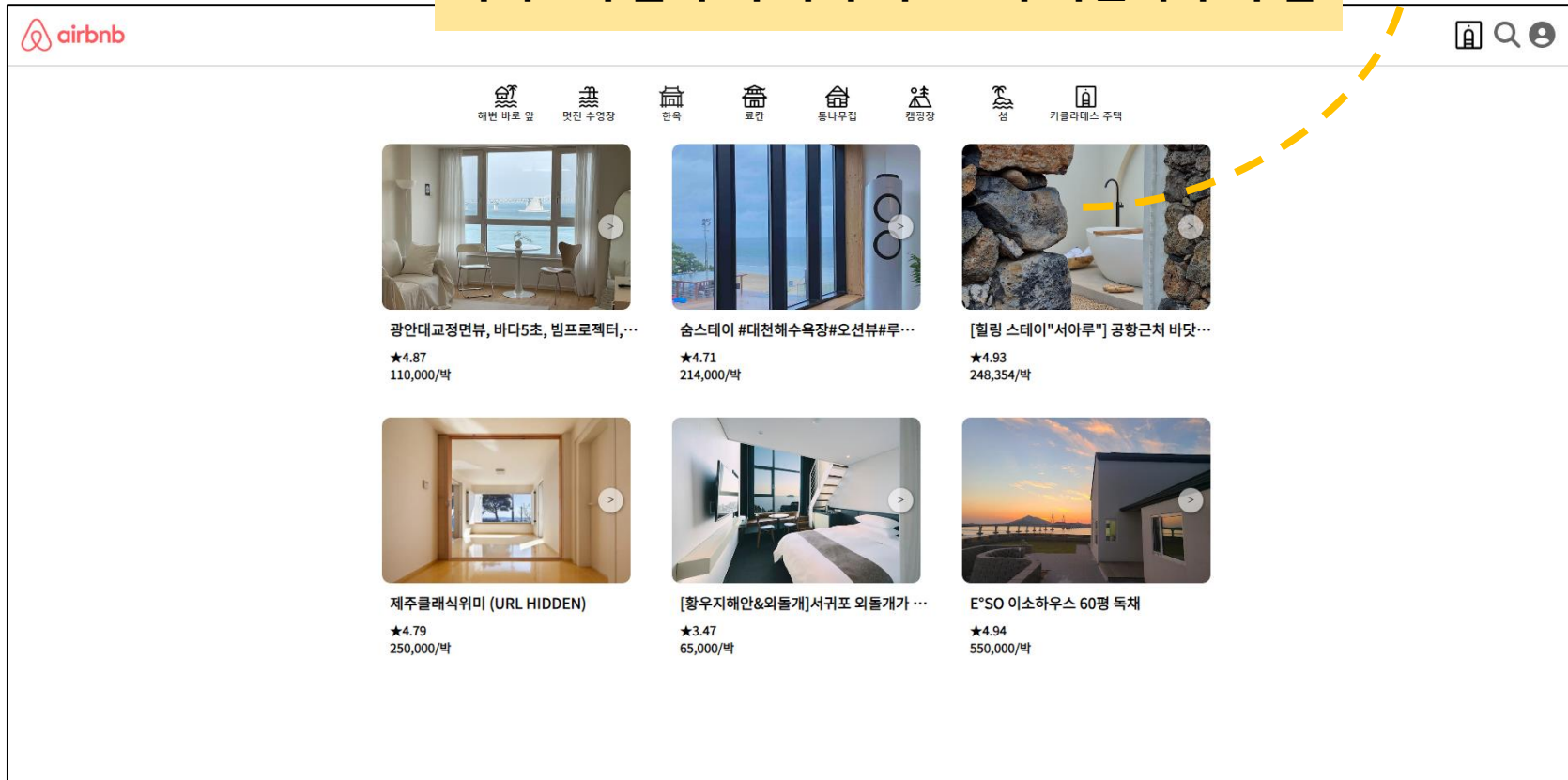
# 01 사이트 설계

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



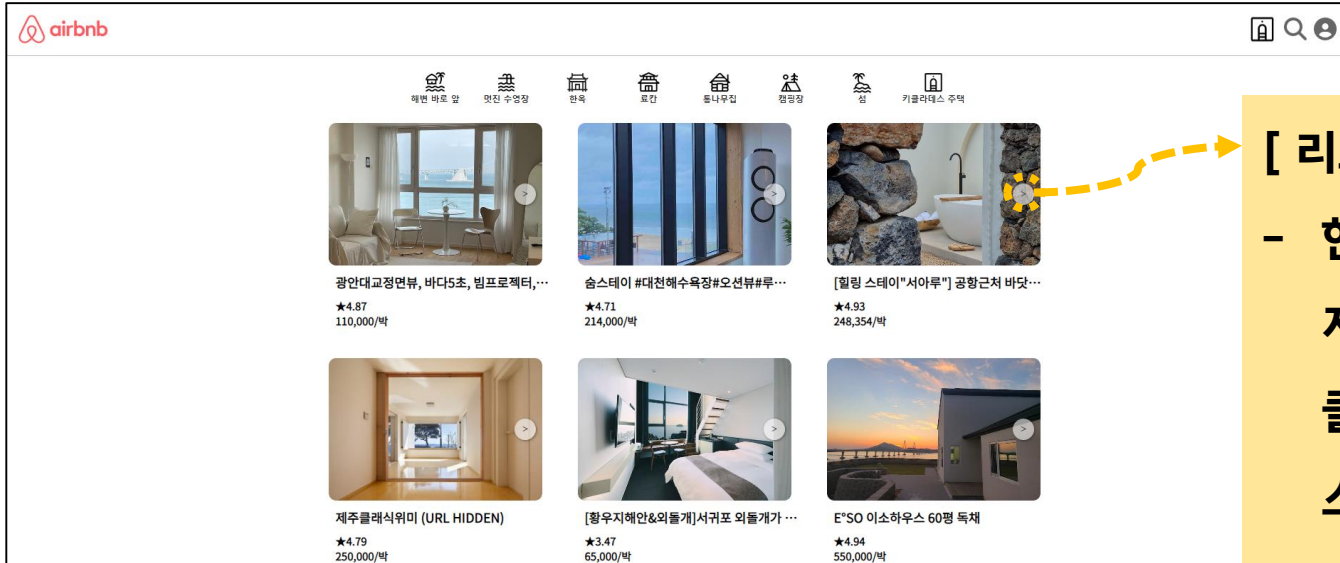
### 03 모바일 설계



GNB메뉴를 세로형 으로 변경

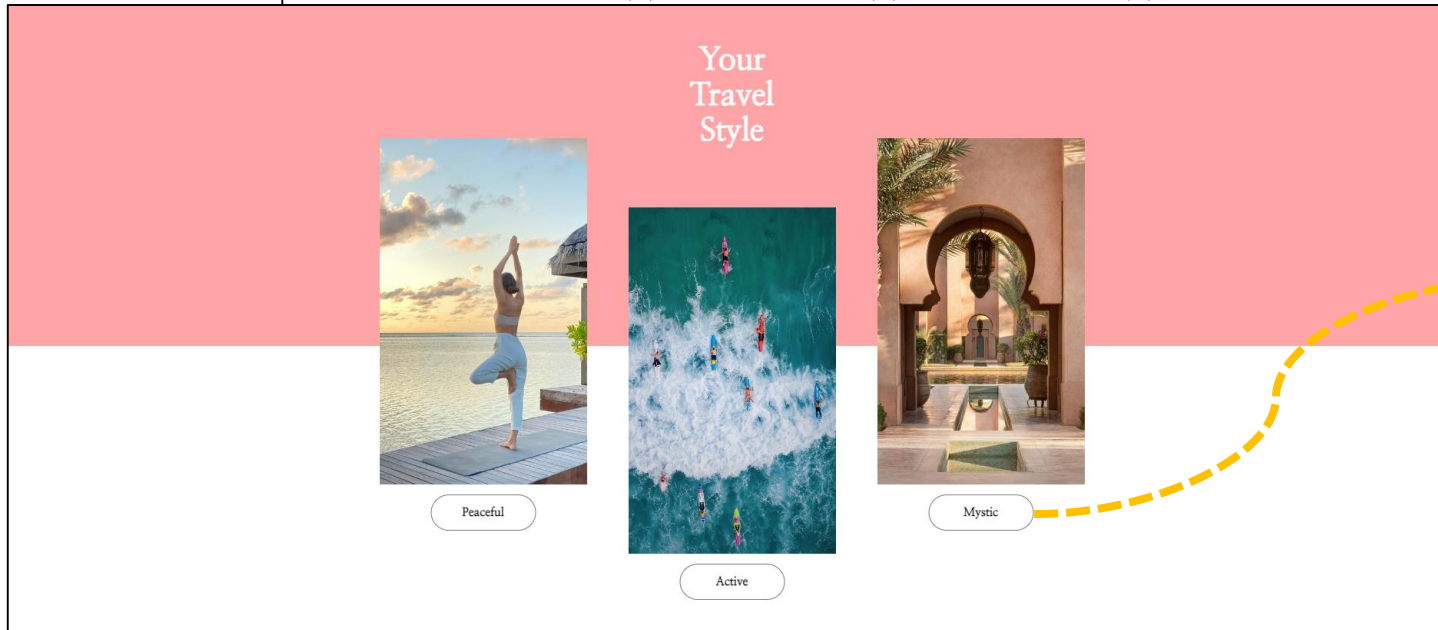
리스트는 화면 크기에 맞춰 그리드 설정

# 개선 항목



## [ 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



## [ 랜덤 카테고리로 이동하는 기능 ]

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



**감사합니다.**