

인터페이스 구현 보고서

소속 팀	B팀	작성자	소연희
인터페이스 명	헤더 / 버거메뉴 / 바텀 메뉴	소요시간	4시간
구현 목표	<p>[헤더] 뒤로가기 버튼, 검색, 버거 메뉴 선택 가능하여 어느 페이지든 사용이 가능하도록 함</p> <p>[바텀 메뉴] 페이지 이동을 위한 메뉴 구성으로 사용자가 손쉽게 사용 가능하도록 함</p>		
구현 내용	<p>[헤더]</p> <p>헤더의 좌측 부분에 들어가는 상황을 3가지로 분류</p> <ol style="list-style-type: none"> 1. Home일 경우에는 로고가 보이게 함 <pre>ex) if(pathname==='/') { head = ; }</pre> <ol style="list-style-type: none"> 2. 뒤로 가기 버튼이 보이는 경우, 해당 버튼을 클릭 시 navigate(-1)을 넣어 뒤로 가기 버튼이 작동되게 함 <pre>ex) else if (slash >= 2) { head = <Arrow className={'header-back'} onClick={()=>navigate(-1)} />; }</pre> <ol style="list-style-type: none"> 3. 필요한 아이콘이 없을 경우는 빈 값을 보냄 <pre>ex) else { head = ""; }</pre> <ul style="list-style-type: none"> - 각 카테고리의 메인 페이지인 경우에는 헤더가 필요없는 상황이므로 hiddenPaths 변수 안에 제거해야 할 경로를 적어 해당 경로가 포함되어 있을 때는 실행되지 않게 작업 - 검색 아이콘과 리스트 아이콘을 넣어 어느 페이지에서든 검색과 버거 메뉴를 클릭하도록 작업 <p>[버거 메뉴]</p> <ul style="list-style-type: none"> - 각 메뉴에 해당하는 페이지로 이동해야하기 때문에 NavLink를 사용 - 버스정보의 경우 타 사이트 링크로 넘어가기 때문에 새 탭으로 넘어가도록 작업 - 버스정보 탭 클릭 시 window.open을 사용('_blank' : 새 탭띄우기, 'noopener, noreferrer' : 보안상 정보 전달 막기) - 헤더에 버거 메뉴가 들어가 있는 형태로 useState를 사용하여 버거 상태 기본값을 false로 둔 후, 버거 메뉴를 클릭했을 때 상태 값이 바뀌게 됨 - 그 값을 버거 컴포넌트에서 받은 후 닫기 버튼을 클릭 시 onClose의 상태값이 false로 변경하여 열린 상태를 관리 <p>[바텀 메뉴]</p> <ul style="list-style-type: none"> - 페이지를 쉽게 이동할 수 있도록 하단에 주요 메뉴들을 넣어 구성 - active 클래스를 넣어 클릭하면 색상이 변경되어 현재 페이지를 인지할 수 있도록 작업 - 헤더와 마찬가지로 바텀 메뉴가 필요 없는 페이지의 경우 hiddenPaths 변수 안에 제거해야 할 경로를 적어 해당 경로가 포함되어있을 때는 실행되지 않게 작업 		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none"> - 헤더와 바텀 메뉴가 존재하는 페이지, 존재하지 않는 페이지로 분류되어 숨겨야 할 페이지를 설정해야 하는 이슈 발생 → hiddenPaths라는 변수를 만들어 숨겨야 할 페이지를 지정하고 해당 변수가 포함되어있을 때 보여주지 말라는 명령으로 처리하였습니다. 또 검색 디테일 페이지에서는 검색어를 입력하게 되면 해당 검색어가 url로 값이 들어가서 location.pathname.startsWith를 추가하여 검색 디테일 페이지에서도 보여지지 않게 처리하였습니다. <pre>ex) const hiddenPaths = ["/login"] const hideHeader = hiddenPaths.includes(location.pathname) location.pathname.startsWith("/search/searchdetail");</pre>		

인터페이스 명	관광지, 쇼핑, 음식점, 축제행사 api 관련	소요시간	50분																																																																												
구현 목표	홈, 트립 카테고리 등에 사용되는 api로 리스트를 보여주기 위함																																																																														
구현 내용	<p>가이드 문서 :</p> <p>https://api.visitjeju.net/file/bbsattachment/filepath/202010/08/8e41d714-aa18-4d37-9e99-7aa5982dd913.pdf</p> <p>1. URL - http://api.visitjeju.net/vsjApi/contents/searchlist 2. apiKey(필수) - 57fd439ed04e408c935a985377cbaa41 3. locale(필수) - kr/en/jp/cn/zh/my 4. category - c1: 관광지 / c2: 쇼핑 / c4: 음식점 / c5: 축제/행사 5. page(선택) - 2 6. cid(선택) - CONT_000000000500349</p> <p>예시 링크 :</p> <p>http://api.visitjeju.net/vsjApi/contents/searchList?apiKey=57fd439ed04e408c935a985377cbaa41&locale=kr&category=c2</p> <p>한 페이지당 출력 개수는 100개, 전체 페이지 수는 카테고리마다 다른 카테고리별 마지막 페이지 + 1을 하면 전체 리스트를 볼 수 있음 ex) currentPage : 23 (전체 리스트 확인 가능)</p> <p>응답메시지 :</p> <table> <tr> <td>result</td><td>결과코드</td><td>00</td><td>결과코드</td></tr> <tr> <td>resultMessage</td><td>결과메시지</td><td>success</td><td>결과메시지</td></tr> <tr> <td>totalCount</td><td>전체 결과 개수</td><td>1152</td><td>해당 조건의 항목의 총 개수</td></tr> <tr> <td>resultCount</td><td>현재 결과 개수</td><td>100</td><td>현재 출력된 결과 개수</td></tr> <tr> <td>pageSize</td><td>페이지당 출력개수</td><td>100</td><td>한 페이지당 출력 개수</td></tr> <tr> <td>pageCount</td><td>전체 페이지수</td><td>12</td><td>전체 페이지 수</td></tr> <tr> <td>currentPage</td><td>현재 페이지</td><td>1</td><td>현재 페이지</td></tr> <tr> <td>items</td><td>출력된 결과 항목</td><td></td><td>출력된 결과 항목 배열</td></tr> <tr> <td>└ contentsid</td><td>콘텐츠 ID</td><td>CONT_000000000500349</td><td>콘텐츠 ID</td></tr> <tr> <td>└ contentscd</td><td>콘텐츠코드</td><td></td><td>콘텐츠코드</td></tr> <tr> <td>└ value</td><td>콘텐츠코드 값</td><td>c1</td><td>콘텐츠코드 값</td></tr> <tr> <td>└ label</td><td>콘텐츠코드 라벨</td><td>관광지</td><td>해당 콘텐츠코드의 라벨</td></tr> <tr> <td>└ title</td><td>콘텐츠명</td><td>성산일출봉(UNESCO 세계자연유산)</td><td>해당 콘텐츠명</td></tr> <tr> <td>└ roadaddress</td><td>도로명주소</td><td>제주특별자치도 서귀포시 성산읍 일출로 284-12</td><td>도로명주소</td></tr> <tr> <td>└ tag</td><td>태그</td><td>일출,오름,경관/포토,부모</td><td>태그</td></tr> <tr> <td>└ introduction</td><td>간단소개</td><td>바다위에 우뚝 솟아난 수성화산·유네스코 세계자연유산, 천연기념물 제420호, 올레1코스</td><td>간단소개</td></tr> <tr> <td>└ repPhoto</td><td>대표 등록사진</td><td></td><td>대표 등록사진</td></tr> <tr> <td>└ photoid</td><td>사진 ID</td><td></td><td>사진 ID</td></tr> <tr> <td>└ imgpath</td><td>일반 이미지경로</td><td>https://api.cdn.visitjeju.net/photomng/imgpath/20181017/c072ee1a-2a02-4be7-b0cd-62f4daf2f847.gif</td><td>일반 이미지경로</td></tr> </table>			result	결과코드	00	결과코드	resultMessage	결과메시지	success	결과메시지	totalCount	전체 결과 개수	1152	해당 조건의 항목의 총 개수	resultCount	현재 결과 개수	100	현재 출력된 결과 개수	pageSize	페이지당 출력개수	100	한 페이지당 출력 개수	pageCount	전체 페이지수	12	전체 페이지 수	currentPage	현재 페이지	1	현재 페이지	items	출력된 결과 항목		출력된 결과 항목 배열	└ contentsid	콘텐츠 ID	CONT_000000000500349	콘텐츠 ID	└ contentscd	콘텐츠코드		콘텐츠코드	└ value	콘텐츠코드 값	c1	콘텐츠코드 값	└ label	콘텐츠코드 라벨	관광지	해당 콘텐츠코드의 라벨	└ title	콘텐츠명	성산일출봉(UNESCO 세계자연유산)	해당 콘텐츠명	└ roadaddress	도로명주소	제주특별자치도 서귀포시 성산읍 일출로 284-12	도로명주소	└ tag	태그	일출,오름,경관/포토,부모	태그	└ introduction	간단소개	바다위에 우뚝 솟아난 수성화산·유네스코 세계자연유산, 천연기념물 제420호, 올레1코스	간단소개	└ repPhoto	대표 등록사진		대표 등록사진	└ photoid	사진 ID		사진 ID	└ imgpath	일반 이미지경로	https://api.cdn.visitjeju.net/photomng/imgpath/20181017/c072ee1a-2a02-4be7-b0cd-62f4daf2f847.gif	일반 이미지경로
result	결과코드	00	결과코드																																																																												
resultMessage	결과메시지	success	결과메시지																																																																												
totalCount	전체 결과 개수	1152	해당 조건의 항목의 총 개수																																																																												
resultCount	현재 결과 개수	100	현재 출력된 결과 개수																																																																												
pageSize	페이지당 출력개수	100	한 페이지당 출력 개수																																																																												
pageCount	전체 페이지수	12	전체 페이지 수																																																																												
currentPage	현재 페이지	1	현재 페이지																																																																												
items	출력된 결과 항목		출력된 결과 항목 배열																																																																												
└ contentsid	콘텐츠 ID	CONT_000000000500349	콘텐츠 ID																																																																												
└ contentscd	콘텐츠코드		콘텐츠코드																																																																												
└ value	콘텐츠코드 값	c1	콘텐츠코드 값																																																																												
└ label	콘텐츠코드 라벨	관광지	해당 콘텐츠코드의 라벨																																																																												
└ title	콘텐츠명	성산일출봉(UNESCO 세계자연유산)	해당 콘텐츠명																																																																												
└ roadaddress	도로명주소	제주특별자치도 서귀포시 성산읍 일출로 284-12	도로명주소																																																																												
└ tag	태그	일출,오름,경관/포토,부모	태그																																																																												
└ introduction	간단소개	바다위에 우뚝 솟아난 수성화산·유네스코 세계자연유산, 천연기념물 제420호, 올레1코스	간단소개																																																																												
└ repPhoto	대표 등록사진		대표 등록사진																																																																												
└ photoid	사진 ID		사진 ID																																																																												
└ imgpath	일반 이미지경로	https://api.cdn.visitjeju.net/photomng/imgpath/20181017/c072ee1a-2a02-4be7-b0cd-62f4daf2f847.gif	일반 이미지경로																																																																												
발생된 이슈 및 해결 방안	-																																																																														


인터페이스 명	관광객 수 api 관련	소요시간	30분									
구현 목표	홈에 사용되는 api로 월별 관광객 통계를 보여주기 위함											
구현 내용	<div>1. URL - http://api.odcloud.kr/api/3083546/v1/uddi:4a4ea6e8-33e6-45c3-9c19-b8efe36ffd3b</div> <div>2. serviceKey(필수) - U0%2FVNFIHUBaYIX80pdY4xpWRnNWmK089qSyEZrhhwobVU599onCKVvNnb0jHHcbQiQ1qcLqZWP21BSzibjqC4Q%3D%3D</div> <div>예시 링크 : http://api.odcloud.kr/api/3083546/v1/uddi:4a4ea6e8-33e6-45c3-9c19-b8efe36ffd3b?serviceKey=U0%2FVNFIHUBaYIX80pdY4xpWRnNWmK089qSyEZrhhwobVU599onCKVvNnb0jHHcbQiQ1qcLqZWP21BSzibjqC4Q%3D%3D</div> <div>응답메시지 :</div> <table><tr><td>currentCount</td><td>현재 개수</td><td>10</td></tr><tr><td>data</td><td>출력된 결과 항목</td><td></td></tr><tr><td>목적별(휴양 및관광)</td><td></td><td>731513</td></tr></table>			currentCount	현재 개수	10	data	출력된 결과 항목		목적별(휴양 및관광)		731513
currentCount	현재 개수	10										
data	출력된 결과 항목											
목적별(휴양 및관광)		731513										
발생된 이슈 및 해결 방안	-											

인터페이스 명	홈	소요시간	24시간
구현 목표	접속 시 가장 먼저 노출되는 화면으로, 제공되는 정보를 간략히 소개하여 사용자가 콘텐츠 구성을 쉽게 파악할 수 있도록 구현		
구현 내용	<p>[섹션 1_메인 슬라이드]</p> <ul style="list-style-type: none"> - SwiperSlide를 사용하여 api 데이터(관광지, 맛집, 축제, 쇼핑) 카테고리에서 콘텐츠를 하나씩 랜덤으로 꺼내어 24시간 동안 화면에 보여줄 데이터를 로컬스토리지에 저장 - 24시간이 지나면 저장되어있던 데이터가 삭제된 후, 새로운 콘텐츠로 변경 - 데이터가 들어오는 시간 동안 로딩 처리를 해두고 데이터가 들어온 후 콘텐츠 클릭 시 디테일 페이지로 이동 <p>[섹션 2_탭 메뉴]</p> <ul style="list-style-type: none"> - 트립의 카테고리인 tour, food, festival, shopping 부분 쉽게 들어갈 수 있도록 메인에 탭을 넣어 구성 - 해당 부분은 고정된 섹션으로 탭 메뉴 클릭 시 해당 카테고리로 이동 <p>[섹션 3_날씨]</p> <ul style="list-style-type: none"> - 데이터가 들어오는 시간 동안 로딩 처리 - 날씨 api 데이터가 들어온 후 현재 기온, 최저 최고 기온, 바람을 보여주어 사용자에게 정보 제공 - 날씨 상태를 이미지로 보여주기 위해 “맑음, 구름많음, 흐림 등등”의 경우 switch를 사용하여 해당 값일 때 이미지를 보여주게 설정 * api는 지현님 파일에 적혀있음, 해당 탭은 클릭 불가하며 날씨 정보 확인용으로 사용됨) <p>[섹션 4_관광지]</p> <ul style="list-style-type: none"> - 관광지 데이터(api)에서 24시간 동안 화면에 보여줄 데이터를 로컬스토리지에 저장 - 저장한 데이터를 화면에 보여주고, 24시간이 지나면 저장되어있던 데이터가 삭제된 후 새로운 콘텐츠로 변경 - 데이터가 들어오는 동안에는 로딩 처리 - 데이터가 들어온 후 펼쳐져있는 첫 번째 콘텐츠를 클릭하게 되면, 바로 해당 디테일 페이지로 넘어감 - 2, 3, 4번째 이미지를 한번 클릭하게 되면 콘텐츠가 펼쳐지고 한 번 더 클릭하게 되면 디테일 		

	<p>페이지로 넘어감</p> <ul style="list-style-type: none">- 전체보기 버튼 클릭 시 관광지 리스트 페이지로 넘어감- 주소에서 제주특별자치도를 제거하고 시부터 보여주어 가독성을 높임 <p>[섹션 5_맛집]</p> <ul style="list-style-type: none">- 맛집 데이터(api)에서 24시간 동안 화면에 보여줄 데이터를 로컬스토리지에 저장- 저장한 데이터를 화면에 보여주고, 24시간이 지나면 저장되어있던 데이터가 삭제된 후 새로운 콘텐츠로 변경- 데이터가 들어오는 동안에는 로딩 처리- 데이터가 들어온 후 콘텐츠를 클릭하게 되면 해당 디테일 페이지로 넘어감- 전체보기 버튼 클릭 시 맛집 리스트 페이지로 넘어감- 주소에서 제주특별자치도를 제거하고 시부터 보여주어 가독성을 높임 <p>[섹션 6_포토]</p> <ul style="list-style-type: none">- 커뮤니티 떠나팁에서 관리자가 업로드한 이미지들을 SwiperSlide를 사용하여 구성- 전체보기 버튼 및 이미지를 클릭 시 로그인 되어있지 않다면 로그인 팝업창 띄움- 로그인 후 해당 이미지 클릭 시 게시물이 있는 디테일 페이지로 이동, 전체보기 버튼 클릭 시 커뮤니티 갤러리 탭으로 이동 <p>[섹션 7_관광객 수]</p> <ul style="list-style-type: none">- 2024년 기준의 관광객 수 데이터(api)를 불러와 기본값을 현재 월로 설정- 홈에 들어오게 되면 자동으로 현재 월이 선택되고 숫자 변경됨- 월을 변경하고 싶을 때, 말풍선 안에 있는 인풋 박스를 선택하여 팝업창 월 선택 후 완료 버튼을 누르게 되면 그 값이 반영되어 관광객 수가 변경- 인풋 박스의 경우 숫자를 직접 입력하는 게 아니라 팝업창이 나와서 버튼으로 선택해야하므로, 입력하는 칸을 readOnly 속성으로 막음- 관광객 수는 선택한 월이 true이고 관광객 수의 length값이 0 이상일 경우, 데이터에서 '구분연월'을 찾아서 선택한 월을 가져옴- 그 데이터 안에 '목적별(휴양및관광)' 카테고리에서 수치값을 꺼내와 넘버 타입으로 보여지게 함- 숫자가 올라가는 애니메이션은 motion.dev 플러그인을 사용해 관광객 수치를 자연스럽게 보여지도록 작업 <p>[하단 탭 버튼]</p> <ul style="list-style-type: none">- 화면상 리스트가 긴 상황이나 현재 페이지가 홈일 경우 상단으로 이동하는 데 불편함을 해소하고자 탭 버튼 생성- 버튼 클릭 시 최상단으로 갈 수 있게 "window.scrollTo({ top: 0, behavior: 'smooth' })"를 사용해 부드럽게 스크롤 되도록 구현 <p>[서버 관련]</p> <ul style="list-style-type: none">- 떠나팁 카테고리의 이미지들을 가져오기 위해 awaitfetch(`\${process.env.REACT_APP_APIURL}/post/images`);로 경로를 잡아주고 filter로 subject가 떠나팁인 것을 찾아옴- 해당 subject에서 이미지와 게시물 아이디 값을 찾아 홈 화면에 이미지를 보여주고, 이미지 클릭시 해당 아이디 값을 가지고 게시물 경로를 찾을 수 있도록 함 <pre>// db 관련 커뮤니티 목록에서 이미지 가져오기 useEffect(()=>{ const fetchPostImg = async ()=>{ const res = await fetch(`\${process.env.REACT_APP_APIURL}/post/images`); if(res.ok){ const result = await res.json(); const imgPosts = result.filter(post=>post.post.subject==='떠나팁') .flatMap(post => post.imageUrl ? [{ imageUrl: post.imageUrl, postId: post.id, post: post }] : []) .slice(0, 5); setImgPost(imgPosts); } } fetchPostImg(); },[])</pre>
발생된 이슈 및 해결 방안	<ul style="list-style-type: none">- 홈 부분의 이슈 및 해결방안은 아래 표에 정리되어 있음

인터페이스 명	홈 섹션 1,4,5 : 슬라이드, 관광지, 맛집 - 하루단위 저장	소요시간	4시간
구현 목표	<p>새로고침 될 때마다 새로운 이미지가 바뀌지 않고, 24시간 동안 같은 이미지를 보여주어 사용자에게 혼동을 주지 않도록 설계함</p>		
구현 내용	<p>[하루단위 랜덤 이미지 생성]</p> <ul style="list-style-type: none"> - 접속한 현재 시간을 가져오고 24시간을 oneDay로 설정하여 데이터를 로컬스토리지에 저장 - 저장 시간이 지났는지 확인하기 위해 변수 isValidTime에 담아 저장 시간이 24시간 이내인지 확인 - 하루가 안 지났다면 저장한 데이터를 보여준 후 return시킴 - 하루가 지났다면 로컬스토리지에서 삭제되도록 작업되고 새로운 데이터를 저장하도록 작업 <pre> if (saveTrip) { try { const parsedTrip = JSON.parse(saveTrip); //저장된 값이 있으면 문자열을 객체로 변경 const { slice, main, createdAt } = parsedTrip; // createdAt 저장 시간이 지났는지 확인하는 용 const isValidTime = createdAt && now - createdAt < oneDay; // 저장시간이 24시간 이내인지 확인 (현재시간-저장시간) if (isValidTime) { // 아직 하루 안 지났음 → 유효 setMainTrip(main); // 랜덤으로 하나 뽑아서 저장 setSelectedTrips(slice); setActiveTrip(slice[0].contentsid); return; } else { // 하루 지남 → 제거 localStorage.removeItem('hometrip'); } } catch (e) { console.error("로컬스토리지 파싱 오류", e); localStorage.removeItem('hometrip'); } } </pre> <ul style="list-style-type: none"> - 새로운 데이터를 저장할 때는 원본 데이터는 유지하고 복사본을 생성하여 랜덤수로 가져옴 - 이때 배열을 랜덤하게 섞기 위해 "[...filterTourList].sort(() => Math.random() - 0.5);" 를 주어 콘텐츠의 앞 뒤 순서도 변경 가능하게 음수값, 양수값을 가지게 작업 - 전체 데이터에서 랜덤으로 하나를 선택하여 선택된 콘텐츠 하나는 메인 슬라이드로 사용됨 - 관광지와 맛집 부분은 변수 copyTourList인 데이터에서 슬라이스로 4개, 6개를 잘라와 useState에 setSelectedTrips에 저장시켜 map을 돌려 구현 <pre> // 여기서 오면 유효한 로컬스토리지 없고 새로 랜덤 생성 const copyTourList = [...filterTourList].sort(() => Math.random() - 0.5); const sliceTrip = copyTourList.slice(0, 4); const mainTripPick = copyTourList[Math.floor(Math.random() * copyTourList.length)]; // 전체 데이터에서 랜덤으로 하나 선택 setMainTrip(mainTripPick); // 랜덤으로 고른걸 메인슬라이드 이미지로 선택 setSelectedTrips(sliceTrip); // 선택한 4개 여행지 저장 setActiveTrip(sliceTrip[0].contentsid); // 첫번째 콘텐츠의 너비는 펼쳐진 상태로 하기 위해 tripData[0].id 적용 const createdAt = new Date().getTime(); //현재 시각ms localStorage.setItem('hometrip', JSON.stringify({slice: sliceTrip, main: mainTripPick, createdAt})); </pre>		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none"> - 가라데이터를 생성 후 쿠키에 하루단위 저장을 했는데 api를 연결하니 쿠키 저장에 오류가 생겨 데이터가 계속 새로고침되는 이슈 발생 → 쿠키로 해결되지 않아 로컬스토리지로 변경하고, 현재 시간을 저장하고 시간이 지나면 삭제하고 다시 랜덤 생성하게 작업하였습니다. ex) 로컬스토리지 저장 : const saveTrip = localStorage.getItem('hometrip'); 로컬스토리지 삭제 : localStorage.removeItem('hometrip'); 		

인터페이스 명	홈 섹션7 : 관광객 수	소요시간	3시간
구현 목표	제주 여행을 오는 관광객 수 통계를 월별로 보여주어 어느 때 관광객이 많이 방문하는지 파악할 수 있도록 함		
구현 내용	<ul style="list-style-type: none"> - 인풋 박스의 월 버튼을 클릭하게 되면 하단에서 팝업창이 올라와 원하는 월을 선택하여 해당 월의 관광객 수 확인 가능 - 팝업창에서 월 버튼을 클릭했을 때, 클릭하지 않았을 때 상태값을 찾음 - 클릭했을 때만 완료버튼이 활성화되게 작업 - 버튼을 클릭하지 않았을 때는 닫기 버튼이나 배경을 클릭하여 팝업창을 닫도록 작업 		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none"> - 팝업창이 열렸을 때, 버튼을 선택하지 않고 완료 버튼을 눌러도 팝업이 닫히게 되는 이슈 → Month.jsx 컴포넌트(월 팝업창)에서 완료 버튼에 값이 없을 경우 return 하라는 조건문을 넣고 버튼을 눌렀을 때만 완료 버튼을 눌러게 변경하였습니다. 또, 선택한 버튼이 없는 경우 완료 버튼을 비활성화하고 버튼을 선택했을 경우 활성화하여 사용자에게 동작 가능 여부를 인지시켰습니다. 		

인터페이스 명	공동 컴포넌트 : PlaceItem	소요시간	40분
구현 목표	리스트를 보여줄 때, 사용되는 컴포넌트로 검색, 트립 디테일, 좋아요 리스트 등에 공용으로 사용하기 위함		
구현 내용	<ul style="list-style-type: none"> - api 정보를 가져와 해당 컴포넌트에 적용시켜 리스트를 보여주는 방식 - 해당 컴포넌트에는 이미지, 타이틀, 주소, 태그, 하트로 구성 - 이미지, 타이틀, 주소, 태그는 프롭스로 보내 사용할 곳에 가서, 데이터 정보로 map을 돌릴 때 받아서 사용 - 여기서 태그는 #으로 되어있어서 #을 기준으로 단어를 나누고, 최대 4개만 보여지게 설정한 후, map으로 공백을 전부 삭제 - 그것을 filter Boolean으로 확인하여 값이 있는 것만 보여지게 설정 - 화면에 보여질 때 단어 사이에 가운데 점을 붙여 노출 ex) {tag}{i !== tagLimit.length - 1 && ' . '} - 하트의 경우, 색상이 들어간 하트, 라인이 있는 하트, 하트가 없는 경우 3가지로 사용 - 변수 heartElement의 기본값을 null로 지정하고, heartType이 red-fill인 경우 LikeRed 컴포넌트에서 받는 liked값이 true이고 heartType이 red-stroke인 경우 LikeRed 컴포넌트에서 받는 liked값이 false로 설정 - 하트를 보여줄 때, 사용할 곳에 가서 프롭스로 heartType을 3가지 형식으로 받음 ex) heartType = "red-fill" / heartType="red-stroke" / heartType="none" - 해당 컴포넌트가 사용되는 곳은 검색 디테일 페이지, 트립 리스트 페이지, 좋아요 페이지, 여행 장소 추가할 수 있는 페이지에서 사용됩니다. <div>  <div> <p>브랜섬홀아시아 아이스링크</p> <p>제주특별자치도 서귀포시 대정읍 글로벌에듀로 234(3번게이트)</p> <p>제주 · 제주도 · 아이스링크 · 스케이트장</p> </div> <div>  0 </div> </div>		
발생된 이슈 및 해결 방안	-		

인터페이스 명	인기 검색어, 인기 태그 api 관련	소요시간	30분																														
구현 목표	검색에 사용되는 api로 인기 검색어와 인기 태그를 보여주고 검색에 용이하게 하기 위함																																
구현 내용	<div>[인기 검색어]</div> <div>1. URL - http://api.odcloud.kr/api/15049993/v1/uddi:04795979-fd51-411e-ad41-95a559b706eb</div> <div>2. serviceKey(필수) - U0/VNFIHUBaYIX80pdY4xpWRnNWmK089qSyEZrhwbobVU599onCKVvNnb0jHHcbQiQ1qcLqZWP21BSzibjqC4Q==</div> <div>[인기 태그]</div> <div>1. URL - http://api.odcloud.kr/api/15049992/v1/uddi:ab253c7b-3e49-40e5-b8bb-f8a925519d1d</div> <div>2. serviceKey(필수) - NICVW7%2BM1Rr%2Fa14RcX1kdhmFrvGCRRK%2Bb6XJJjagnrepPxv%2B0cjSfIrkZ9YEKbQKG6wBcoUNyMLi32eC%2F%2Fh81fg%3D%3D</div> <div>예시 링크 :</div> <div>http://api.odcloud.kr/api/15049992/v1/uddi:ab253c7b-3e49-40e5-b8bb-f8a925519d1d?serviceKey=NICVW7%2BM1Rr%2Fa14RcX1kdhmFrvGCRRK%2Bb6XJJjagnrepPxv%2B0cjSfIrkZ9YEKbQKG6wBcoUNyMLi32eC%2F%2Fh81fg%3D%3D</div> <div>인기 검색어 응답메시지 :</div> <table><tr><td>currentCount</td><td>현재 개수</td><td>10</td></tr><tr><td>data</td><td>출력된 결과 항목</td><td></td></tr><tr><td>└ 단어</td><td></td><td>제주</td></tr><tr><td>└ 일련번호</td><td></td><td>101</td></tr><tr><td>└ 정렬순서</td><td></td><td>1</td></tr></table> <div>인기 태그 응답메시지 :</div> <table><tr><td>currentCount</td><td>현재 개수</td><td>10</td></tr><tr><td>data</td><td>출력된 결과 항목</td><td></td></tr><tr><td>└ 일련번호</td><td></td><td>133640</td></tr><tr><td>└ 정렬순서</td><td></td><td>1</td></tr><tr><td>└ 태그명</td><td></td><td>자연경관</td></tr></table>			currentCount	현재 개수	10	data	출력된 결과 항목		└ 단어		제주	└ 일련번호		101	└ 정렬순서		1	currentCount	현재 개수	10	data	출력된 결과 항목		└ 일련번호		133640	└ 정렬순서		1	└ 태그명		자연경관
currentCount	현재 개수	10																															
data	출력된 결과 항목																																
└ 단어		제주																															
└ 일련번호		101																															
└ 정렬순서		1																															
currentCount	현재 개수	10																															
data	출력된 결과 항목																																
└ 일련번호		133640																															
└ 정렬순서		1																															
└ 태그명		자연경관																															
발생된 이슈 및 해결 방안	-																																

인터페이스 명	검색 : 메인 페이지	소요시간	2시간
구현 목표	api 정보들을 쉽게 찾을 수 있는 페이지로 인기 검색어와 인기 태그들을 보여줌		
구현 내용	<ul style="list-style-type: none"> - SearchBar 컴포넌트를 사용하여 해당 페이지에 맞게 검색창으로 사용 - 입력할 때마다 searchInput의 값을 업데이트하도록 설정 - searchInput의 값이 공백이 아닐 경우 변수 wordClick에 검색어가 들어가도록 useNavigate를 사용하여 경로를 설정 - 해당 경로 안에 사용된 “encodeURIComponent(word)”는 띄어쓰기나 특수문자가 있을 경우 대비하여 문자열을 안전하게 인코딩해주어 서버가 잘못 해석하거나 검색 결과가 안 뜨는 경우를 방지 - 인기 검색어, 인기 태그의 정보들을 api로 불러오는 동안에 대비하여 로딩 처리를 해두고 데이터가 들어올 수 있도록 처리 - api로 불러온 인기 검색어, 인기 태그들을 클릭했을 때에도 wordClick 함수를 실행시켜 사용자가 편리하게 검색할 수 있도록 구현 		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none"> - 인기 검색어 중 띄어쓰기가 있는 단어일 경우 검색이 되지 않는 이슈 → 검색어에 띄어쓰기가 있는 경우 api 데이터를 불러올 때 해당 데이터를 map으로 돌려 “...item, 단어: item.단어.replace(/Ws/g, ' ')”을 사용하여 띄어쓰기를 빈 공백으로 붙여서 수정하여 검색에 용이하게 작업하였습니다. 		

인터페이스 명	검색 : 디테일 페이지	소요시간	3시간
구현 목표	<p>해당 검색어에 맞는 콘텐츠들을 보여줌</p> <p>검색 디테일 페이지의 상단에 사용자가 검색한 단어가 포함되어있는 콘텐츠들의 총 개수를 표기하고 카테고리별로 콘텐츠를 분류하여 사용자들이 편리하게 사용할 수 있도록 함</p>		
구현 내용	<p>[검색창 하단 카테고리]</p> <ul style="list-style-type: none"> - 검색어를 입력한 후, 검색창 아래 h3 태그에 해당 검색어와 관련된 콘텐츠의 총 개수를 데이터의 length값을 가지고 와 화면에 표시 - 검색어를 카테고리별(관광지, 맛집, 축제&행사, 쇼핑)로 분류하여 원하는 카테고리의 콘텐츠를 확인할 수 있도록 구현 - allData의 변수에 카테고리 데이터를 전부 담아, 검색어가 존재하면 입력값을 word로 설정 - 입력된 값이 콘텐츠의 제목, 주소, 태그에 걸리게 되면 결과에 포함시켜 filteredData에 저장 - 저장된 값을 카테고리별로 나눠서 보여주기 위해 라벨값이 관광지, 음식점, 쇼핑, 축제/행사일 경우를 찾아 분류 - 사용자들이 편하게 사용하기 위해 상단에 카테고리별로 나눈 부분을 pc버전에서는 fixed로, 모바일 버전에서는 fixed를 해제해 화면 공간을 더 넓게 활용할 수 있도록 구성 <p>[검색 결과 리스트 수]</p> <ul style="list-style-type: none"> - 검색한 리스트의 개수가 많을 경우를 대비하여 한 번에 보여지는 리스트의 개수를 30개로 제한 - 스크롤 상단 값, 브라우저의 화면 높이, 전체 페이지 높이를 계산하여 화면의 가장 하단 지점에 도달했을 때 다음 리스트가 보여지도록 함 		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none"> - 검색 디테일 페이지에서 검색어를 수정할 때, 검색창 아래에 있는 h3 태그에 실시간으로 입력된 값을 받아 반영되는 이슈 → useState 상태를 하나 추가하여 실시간으로 반영되는 값을 막고, 값을 입력하고 검색하게 되면 그 단어를 저장시켜 value값을 변경시켰습니다. 		

인터페이스 명	트립 : 메인 페이지	소요시간	2시간
구현 목표	관광지, 맛집, 축제/행사, 쇼핑 부분을 카테고리별로 분류하여 사용자들이 원하는 콘텐츠를 확인할 수 있도록 트립의 메인 페이지를 구성함		
구현 내용	<p>[하루단위 랜덤 이미지 생성]</p> <ul style="list-style-type: none">- 카테고리당 랜덤으로 게시물 5개씩 꺼내오는 방식으로 앞에 홈에서 했던 방식과 같이 24시간을 oneDay로 설정하여 데이터를 로컬스토리지에 저장- 저장 시간이 지났는지 확인하기 위해 변수 isValidTime에서 저장 시간이 24시간 이내인지 확인- 하루가 안 지났다면 저장한 데이터를 보여주고 다시 return시킴- 하루가 지났다면 로컬스토리지에서 삭제되도록 작업- 새로운 데이터를 저장할 때는 원본 데이터를 유지하고 복사본을 생성하여 랜덤수를 가지고 옴- 배열을 랜덤하게 섞기 위해 "[...filterTourList].sort(() => Math.random() - 0.5);"를 사용하여 콘텐츠를 변수에 저장하고 slice로 해당 변수 5개를 가지고 옴- 잘라온 데이터를 setTripTour에 담아 현재 시간을 인지하게 한 후, 데이터를 SwiperSlide로 구성하여 화면에 노출		
발생된 이슈 및 해결 방안	-		

인터페이스 명	트립 : 리스트 페이지	소요시간	5시간
구현 목표	사용자가 원하는 카테고리를 선택한 후, 그에 맞는 콘텐츠 정보 전달을 위함		
구현 내용	<p>[상단 메인 이미지]</p> <ul style="list-style-type: none"> - 홈이나 트립 메인 페이지에서 리스트 페이지로 넘어왔을 때, 사용자가 선택한 카테고리를 상단에 보여줌 - 카테고리별 대표 이미지를 3장씩 저장하여 24시간 동안 같은 이미지가 보여지게 로컬스토리지에 저장(위에 설명한 방식과 같음) <p>[타이틀, 서브타이틀, 필터 부분]</p> <ul style="list-style-type: none"> - 트립 메인 페이지에 있었던 타이틀, 서브타이틀도 함께 보여주고 우측에는 필터를 사용하여 오름차순, 내림차순, 좋아요순으로 사용자가 선택해서 볼 수 있도록 구성 - 필터는 전체 데이터를 가지고 와서 오름차순, 내림차순일 때 문자열을 비교해서 정렬 - 좋아요순은 아이템의 좋아요 정보를 담기 위한 useState로 likeData, setLikeData를 만든 후 좋아요 총 개수를 내림차순으로 정렬 <p>[콘텐츠 리스트 보여주기]</p> <ul style="list-style-type: none"> - 콘텐츠의 정보를 보여주기 위해 api 정보를 type별로 불러온 후 필터 처리한 값인 변수 getFilterData을 토대로 map을 활용하여 적용시킴 - 이때 데이터가 들어오는 시간을 고려하여 상향 연산자를 통해 로딩 처리함 - 한 번에 보여지는 리스트의 개수를 30개로 지정하기 위해, 스크롤의 상단 값, 브라우저의 화면 높이 값, 전체 페이지 높이를 확인하고 가장 아래 위치 지정에 도달했을 때, 다음 리스트 30개를 보여주기 위해 useState로 상태 값을 만들어 기본값 30을 지정하고 map 적용 시 .slice(0, listCount)를 사용하여 무한 스크롤 방식을 구현함 <p>[서버 관련 : 좋아요순 필터 설정 부분]</p> <ul style="list-style-type: none"> - 사용자가 좋아요순 필터를 선택하게 되면, 각 게시물의 좋아요 수를 가져오기 위해 서버에 게시물 ID를 전송(/triplike/count-mult 엔드포인트 사용) - 서버로부터 받은 postId와 count 데이터를 likeMap 객체에 postId: count 형식으로 저장하고, setLikeData를 통해 상태 반영 - 정렬 시 likeData[a.contentsid] 0; 과 같이 데이터가 없을 경우를 대비해 기본값 0을 설정하고 bLikes - aLikes 방식으로 내림차순 정렬 <p>[서버 관련 : 리스트 좋아요 부분]</p> <ul style="list-style-type: none"> - 각 콘텐츠 우측의 하트 아이콘은 해당 게시물의 총 좋아요 수와 현재 사용자가 좋아요를 눌렀는지 확인 - 사용자가 로그인된 상태라면 “/triplike/liked-posts?userId=\${userId}” 요청을 통해 현재 사용자가 좋아요한 게시물의 ID를 받아옴 - 받아온 ID 리스트(likedPostIds)는 setLikedPosts를 통해 상태로 저장 - 해당 ID들을 기준으로 좋아요 상태를 렌더링함 - 좋아요 수를 보여주기 위해 fetchLikeData 함수가 실행되어, 서버에 likedPostIds를 전송하고 좋아요 수 데이터를 받아 likedData 상태로 저장 ex) likeData = { 41321516: 23 } - likeCount = likeData[contentsId] 0 코드를 사용하여 콘텐츠별 좋아요 숫자를 표시 		
발생된 이슈 및 해결 방안	-		

인터페이스 명	트립 : 디테일 페이지	소요시간	5시간
구현 목표	<p>사용자가 선택해서 들어온 콘텐츠의 상세내용을 보여줌</p> <p>원하는 게시물일 경우 좋아요 기능을 활용하여 저장시킬 수 있음</p>		
구현 내용	<p>- 해당 콘텐츠의 타이틀, 주소, 소개, 전화번호, 태그들을 보여주며 정보를 제공하고, 인근의 장소들을 추천하여 다른 콘텐츠를 손쉽게 확인할 수 있도록 구성</p> <p>[전화번호]</p> <p>- 전화번호가 없는 경우, 지역번호가 없는 경우 등을 대비하여 대체 문구를 설정</p> <p>- 올바른 3파트 형태 `\${parts[0]}-\${parts[1]}-\${parts[2]}`를 잡아주고 전화번호가 2파트만 있는 경우 앞에 지역번호를 붙여서 보여줌</p> <p>- 만약에 하이픈 없이 숫자만 있을 경우, 해당 파트 뒤에 하이픈을 넣을 수 있도록 함</p> <p>[태그]</p> <p>- api를 확인했을 때, 콘텐츠의 태그 부분에는 #이 붙어있어 #을 쉼표로 통일</p> <p>- 쉼표로 문자열을 분리하여 앞 뒤와 빈문자가 있을 경우 해당 부분을 제거</p> <p>- 그 후 태그 컴포넌트를 사용하여 map을 돌려 작업</p> <p>[주변 콘텐츠]</p> <p>- “이 곳은 어때?” 부분의 슬라이드 형식의 콘텐츠는 선택해서 들어온 콘텐츠의 지역을 추출하여 인근에 위치해 있는 곳 중 가볼만한 곳을 보여주는 파트</p> <p>- 선택한 콘텐츠의 주소 중 00시 00읍/면/동 까지 잘라내어 같은 지역의 콘텐츠 목록을 추출</p> <p>- 해당 타입이 tour일 경우 tourlist에서만 콘텐츠를 뽑아내는 방식으로 선택한 콘텐츠 외 다른 콘텐츠들이 보여질 수 있도록 함</p> <p>[이전/이후 버튼]</p> <p>- 트립 디테일 페이지를 나가지 않아도 리스트에 있던 콘텐츠를 확인할 수 있는 버튼</p> <p>- 이전, 이후 버튼의 경우 리스트 페이지에서 사용했던 state로 filterList를 가지고 와 해당 필터 값이 적용된 상태에서 버튼을 클릭했을 때, 이전 이후로 이동되게 함</p> <p>[좋아요]</p> <p>- 좋아요 기능은 로그인을 했을 때만 사용 가능하며, 로그인하지 않았을 때 눌렀을 경우 로그인이 필요한 서비스라는 팝업창을 보여주게 되어 로그인 유도</p> <p>- 로그인 후 좋아요 버튼을 눌렀을 경우, 현재 상태와 반대값으로 변경시켜줌</p> <p>[서버 관련 : 콘텐츠 좋아요 기능]</p> <p>- 로그인 여부를 확인한 후, 서버에 현재 게시물 ID와 사용자 ID, 좋아요 상태를 함께 POST방식으로 전송 ex) const triplike = { userId, postId, liked: !liked };</p> <p>- 요청 성공 시, liked 상태값을 받아 setLiked(result.liked)로 화면에 반영</p> <p>- 좋아요를 누른 게시물을 클릭했을 때, 좋아요 눌렀는지 확인하기 위해 GET으로 요청 ex) /triplike?userId=\${userId}&postId=\${postId}</p> <p>- 서버 응답에서 result.liked 값을 받아 setLiked(result.liked)로 좋아요 상태를 초기화</p> <p>- 페이지가 로딩중일 경우 실행되지 않도록 하여, 렌더링 중 오류 방지</p> <p>- 이를 컴포넌트에서 liked상태와 likeClick 함수가 프롭스로 전달됨</p>		
발생된 이슈 및 해결 방안	<p>- 리스트 페이지에서 필터링을 선택한 후, 디테일 페이지의 하단에 이전, 이후 버튼을 클릭하게 되면 필터링된 값이 저장되지 않아 기본인 오류차순으로 다시 돌아가는 이슈 → TripList페이지에서 navigate함수로 이동하면서 state에 filterList 담아 필터링된 리스트를 보낸 후, TripDetail 페이지에서 useLocation 훅을 사용하여 location.state 안에 들어있는 filterList의 값을 가지고 와 필터링된 값이 적용되게 하였습니다.</p>		