레시피 공유 웹 사이트

리액트 프로젝트 포트폴리오

교육 센터 : 모블 평생교육원

학교 : 나사렛대학교

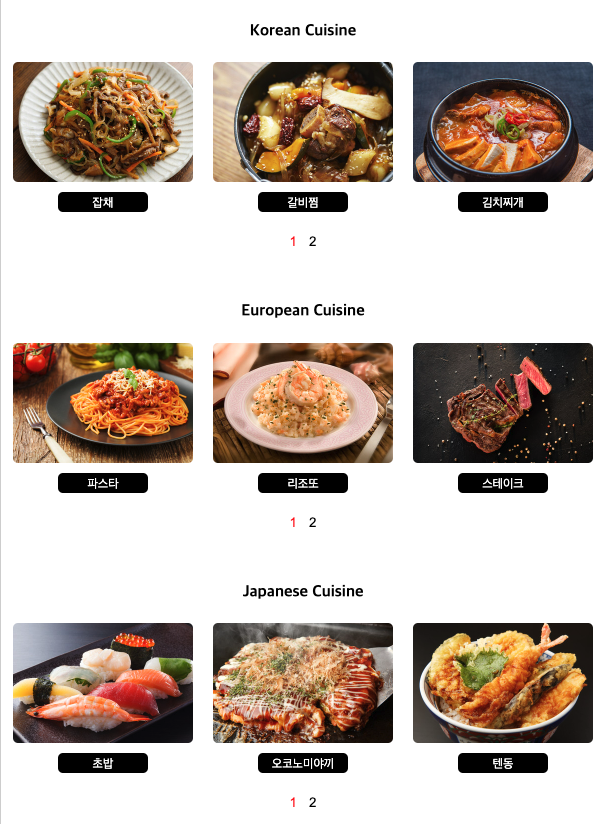
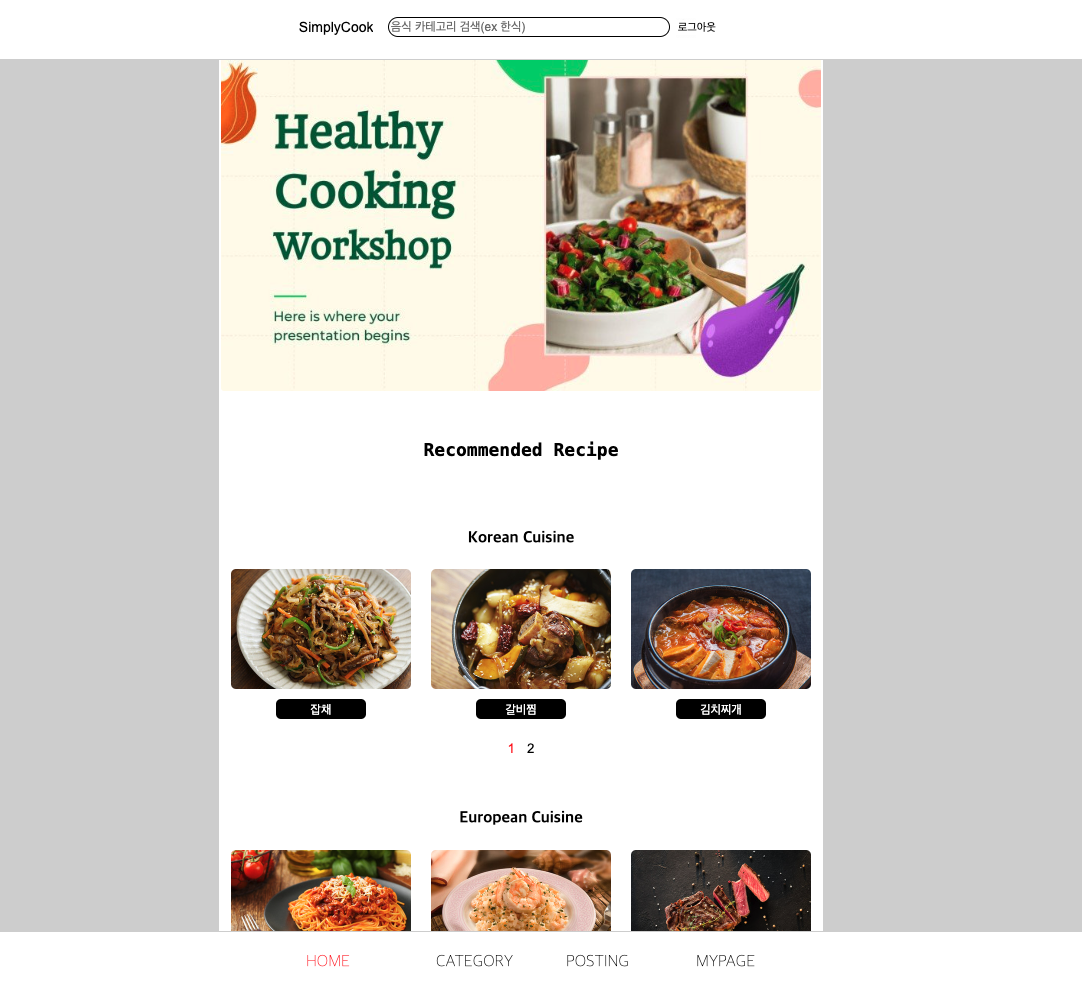
교육생 : 강민준

화면 설계

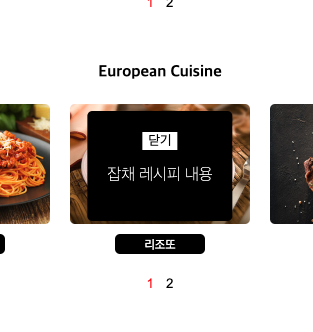
역할: 메인 페이지 레이아웃과 Header, Footer부분 화면 설계

화면 구성

1. 전체 2. 메뉴 레시피 팝업 페이지 부분



3. 팝업으로 보여지는 레시피

 메뉴를 클릭하면 해당 레시피 내용 팝업으로 출력.

1. 레이아웃 나누기 및 설계방법

- header 부분과 footer 부분을 상단, 하단에 positon: fixed CSS 속성으로 고정시킴.

- header와 footer는 다른 페이지에 보여지도록 해야하기에 z-index CSS 속성으로 화면에 가장 먼저 노출되도록 설정함.

- 중앙 영역만 스크롤이 내려가도록 구현.

- 그리드를 사용하여 메뉴들을 카테고리에 따라 사진을 배치 (사진, 팝업 버튼, 팝업 내용 데이터를 넣는 방식으로 함.)

- 팝업 레시피 메뉴 영역을 배열을 이용해 나누어 페이지화 시킴.

2. 추가 기능

- 검색 기능 구현 : 메뉴를 검색하면 필터링 (프로젝트 당시에 구현을 마무리 하지 못함.)



3. 데이터 관리

- 데이터 저장. (웹 브라우저 local storage에 저장하는 방식.)

- 데이터 호출. (JSON을 이용해 local storage에 저장된 정보를 불러오는 방식.)

4. 코드 구성

|  |  |  |
| --- | --- | --- |
| **폴더명** | **파일명** | **설명** |
| page | Corepage.jsx | 중앙영역 코드 |
| Search.jsx | header(검색창과 로그인) 영역 코드 |
| UnderMenu.jsx | 다른 페이지로 넘어갈 수 있는 footer 영역 코드 |

화면 구현

Corepage.jsx

- 카테고리 별 그리드 데이터 넣는 코드부분

// 양식 그리드 데이터

const EuropeanGridData = [

{

image: '/image/grid/grid7.png',

buttonText: '파스타',

popupContent: '파스타 레시피 내용',

},

{

image: '/image/grid/grid8.png',

buttonText: '리조또',

popupContent: '리조또 레시피 내용',

},

{

image: '/image/grid/grid9.png',

buttonText: '스테이크',

popupContent: '스테이크 레시피 내용',

},

{

image: '/image/grid/grid10.png',

buttonText: '감바스',

popupContent: '감바스 레시피 내용',

},

{

image: '/image/grid/grid12.png',

buttonText: '후렌치후라이',

popupContent: '후렌치후라이 레시피 내용',

},

];

{

image: '/image/grid/grid5.png',

buttonText: '비빔밥',

popupContent: '비빔밥 레시피 내용',

},

{

image: '/image/grid/grid6.png',

buttonText: '떡볶이',

popupContent: '떡볶이 레시피 내용',

},

];

// 한식 그리드 데이터

const KoreanGridData = [

{

image: '/image/grid/grid1.png',

buttonText: '잡채',

popupContent: '잡채 레시피 내용',

},

{

image: '/image/grid/grid2.png',

buttonText: '갈비찜',

popupContent: '갈비찜 레시피 내용',

},

{

image: '/image/grid/grid3.png',

buttonText: '김치찌개',

popupContent: '김치찌개 레시피 내용',

},

{

image: '/image/grid/grid4.png',

buttonText: '된장찌개',

popupContent: '된장찌개 레시피 내용',

},

{

image: '/image/grid/grid6.png',

buttonText: '떡볶이',

popupContent: '떡볶이 레시피 내용',

},

];

팝업 구현 코드

// 페이지당 그리드 아이템 개수

const itemsPerPage = 3;

// 현재 페이지의 한식 그리드 아이템

const currentKoreanGridData = KoreanGridData.slice(

currentKoreanPage \* itemsPerPage,

(currentKoreanPage + 1) \* itemsPerPage

);

// 현재 페이지의 양식 그리드 아이템

const currentEuropeanGridData = EuropeanGridData.slice(

currentEuropeanPage \* itemsPerPage,

(currentEuropeanPage + 1) \* itemsPerPage

);

// 현재 페이지의 일식 그리드 아이템

const currentJapaneseGridData = JapaneseGridData.slice(

currentJapanesePage \* itemsPerPage,

(currentJapanesePage + 1) \* itemsPerPage

);

// 팝업 열기 함수

const handleOpenPopup = (content) => {

setPopupContent(content);

setIsPopupOpen(true);

};

// 팝업 닫기 함수

const handleClosePopup = () => {

setPopupContent('');

setIsPopupOpen(false);

};

그리드 구현 및 페이지 버튼 구현 코드

{/\* 유럽 음식 그리드 \*/}

<div style={CorepageStyles.RecommendWrapper}>

<h4>European Cuisine</h4>

</div>

<div style={CorepageStyles.gridContainer}>

{/\* 모듈 구현 \*/}

{currentEuropeanGridData.map((data, index) => (

<div key={index} style={CorepageStyles.gridItem}>

<img src={data.image} alt={`grid${index}`} style={CorepageStyles.gridImage}/>

<button type='button' style={CorepageStyles.button} onClick={() => handleOpenPopup(data.popupContent)}>

{data.buttonText}

</button>

</div>

))}

</div>

{/\* 양식 그리드 페이지 버튼 \*/}

<div>

{Array.from({ length: Math.ceil(EuropeanGridData.length / itemsPerPage) }, (\_, i) => ( //배열을 이용한 페이지 구현

<button key={i} onClick={() => handleEuropeanPageChange(i)} style={{background:'white', border: 'none', color: currentEuropeanPage === i ? 'red' : 'black'}}>

{i + 1}

</button>

))}

</div>

Search.jsx (검색과 로그인이 가능한 header 영역 코드)

{/\* 검색창 \*/}

<input

type="text"

value={query}

onChange={(e) => setQuery(e.target.value)}

onKeyDown={handleKeyDown} // 엔터 키 감지

placeholder="음식 카테고리 검색(ex 한식)"

style={{ width: "30%", height: "40%", fontSize: 12, borderRadius: "10px", border: "1px solid black"}}

onClick={handleSearchBarClick}

/>

{/\* 검색 버튼 \*/}

<button

style={searchStyles.LoginBtn}

onClick={handleSearch}

>

<img src='/searchicon.gif' style={{width:'20px', height:'20px'}}/>

</button>

{/\* 로그인 버튼 \*/}

<button

style={LoginbtnStyles.LoginBtn}

onClick={handleLoginClick}

>

{isLoggedIn ? "로그아웃" : "로그인" }

</button>

</div>

{/\* 검색 필터링 \*/}

<ul style={{ overflowY: "scroll", width: "10%", height: "50%", textAlign: "left", listStyle: "none", padding: 0, border:"none"}}>

{filterCategory(searchResults).map((item, index) => ( // 필터링

<li key={index}>{item}</li>

))}

</ul>

UnderMenu.jsx(다른 페이지로 이동할 수 있는 Footer 영역 코드)

return (

<div style={mainStyle}>

<Outlet/>

<div style={UnderMenuStyles.MenuContainer}>

<div style={{ display: 'flex', justifyContent: 'center', alignItems: 'center', marginRight: '5px', marginLeft:'45px'}}>

<NavLink to="/" style={selectedButton === 'HOME' ? UnderMenuStyles.selectedIconStyle : UnderMenuStyles.IconStyle} onClick={() => setSelectedButton('HOME')}>

HOME

</NavLink>

<NavLink to="/foodcategory" style={selectedButton === 'CATEGORY' ? UnderMenuStyles.selectedIconStyle : UnderMenuStyles.IconStyle} onClick={() => setSelectedButton('CATEGORY')}>

CATEGORY

</NavLink>

<NavLink to="/Writingpage" style={selectedButton === 'POSTING' ? UnderMenuStyles.selectedIconStyle : UnderMenuStyles.IconStyle} onClick={() => setSelectedButton('POSTING')}>

POSTING

</NavLink>

<NavLink to="/mypage" style={selectedButton === 'MYPAGE' ? UnderMenuStyles.selectedIconStyle : UnderMenuStyles.IconStyle} onClick={() => setSelectedButton('MYPAGE')}>

MYPAGE

</NavLink>

</div>

</div>

</div>

);

}