



---

# 웹프로그래밍활용 기말 과제

---

React 활용 웹페이지 제작



2025.12.07



# 목차

<b>1. 개요 .....</b>	<b>1</b>
1-1. 과제 요구사항 .....	1
<b>2. 주제 .....</b>	<b>2</b>
2-1. 주제 .....	2
2-2. 주제 선정 이유 .....	2
<b>3. 구현 내용 .....</b>	<b>3</b>
3-1. User Home Page.....	3
3-2. Dictionary Page .....	4
3-3. Fish Info Page .....	6
3-4. Match Page.....	7
3-5. About Page.....	8
<b>4. 고찰 .....</b>	<b>10</b>

# 1. 과제 개요

## 1-1. 과제 요구사항

---

### 1-1-1. 주제

---

1 학기 웹프로그래밍기초 수업시간에 제출했던 홈페이지의 대표 페이지 2 페이지 이상을 React로 작성할 것. (함수형 컴포넌트 사용)

### 1-1-2. 내용

---

보고서 형식으로 작성하며 아래의 내용을 포함할 것.

- 표지
- 주제선정이유
- 구현페이지 화면 캡처 및 설명 (기술, UI 편의성)

### 1-1-3. 제출 방식

---

수업시간에 사용하는 GitHub Repository에 ★ 기말고사과제 ★ 커밋메시지 작성하여 업로드

## 2. 주제

### 2-1. 주제

---



Aqua Mate 는 물고기들의 합사 궁합을 확인할 수 있는 사이트입니다. 물고기를 키우기 시작한 사람이라면 한 번쯤 고민해 봤을 고민은 “이 물고기와 이 물고기를 한 어항에 키워도 될까?”입니다. 이 사이트는 그런 고민을 덜어드리기 위해 만들어졌습니다. Aqua Mate 는 각종 물고기 사전을 통해 각종 물고기의 기본적인 정보와 사육 정보를 확인할 수 있으며, 성격, 수질 조건, 크기, 활동 영역 등을 기반으로 합사 가능 여부를 알려주는 물생활 도우미입니다.

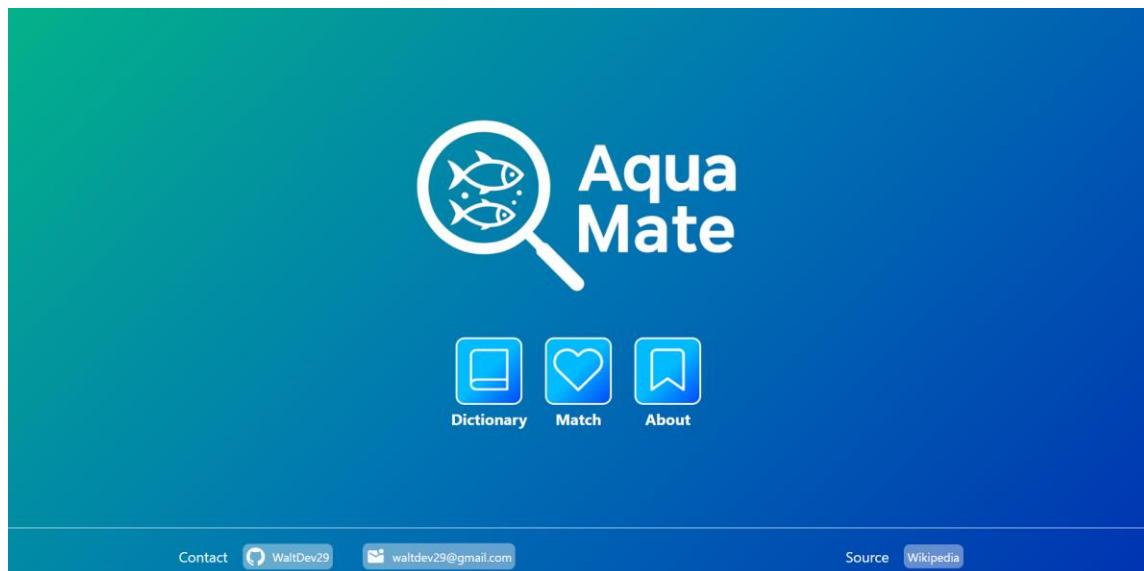
### 2-2. 주제 선정 이유

---

반려동물로 물고기를 키우는 것을 ‘물생활’이라고 합니다. 이 물생활에 처음 도전하는 사람들에게 물고기에 대한 상세한 정보와 사육 방법은 낯설고 어렵게 다가올 수 있습니다. 또한 다양한 물고기를 키우는 사람들도 새로운 물고기를 키우려 할 때 다른 물고기들과 함께 키워도 될지 알기 어려울 수 있습니다. 물고기에 대한 정보에 쉽게 접근하고 어떤 물고기와 함께 키울 수 있는지 확인할 수 있는 사이트를 고안하게 되었습니다.

### 3. 구현 내용

#### 3-1. User Home Page



[User Home Page]

##### 3-1-1. 페이지 설명

User Home Page 는 사용자가 Aqua Mate 웹사이트 접속 시 표시되는 첫 페이지입니다. 화면 중앙에 Aqua Mate 의 로고와 내비게이션 버튼을 배치하였으며, 내비게이션 버튼을 클릭하여 물고기 사전, 물고기 궁합 확인, 웹사이트 소개 페이지로 이동할 수 있습니다. 페이지 최하단에는 Footer 를 배치하여 제작자 정보와 데이터 출처를 명시하였으며, 라벨을 클릭하여 해당 웹사이트로 이동할 수 있습니다.

##### 3-1-2. 기술 설명

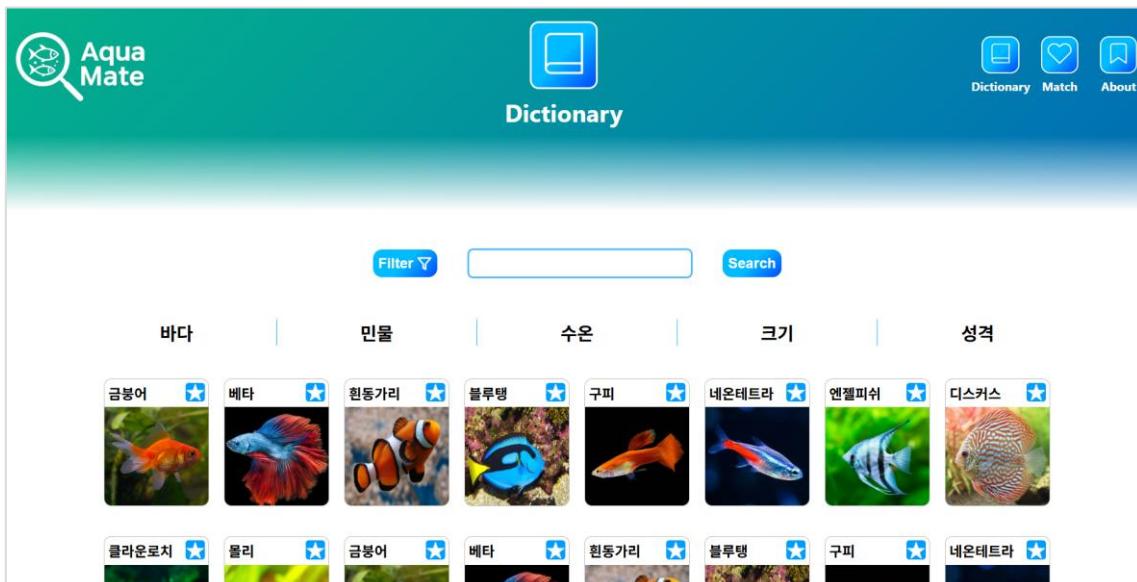
Layout.jsx 컴포넌트를 만들어 Header 와 Footer 를 만들고 props.children 을 사용하여 Layout.jsx 컴포넌트가 하위 컴포넌트를 포함하는 컴포넌트 합성을 활용하였습니다. 본 웹사이트 전체에 Layout.jsx 컴포넌트를 사용하였으며, User Home Page 에서는 Header 가 렌더링 되지 않게 설정하였습니다.

NavBtn.jsx 컴포넌트를 만들어 내비게이션 버튼들을 만들었습니다. Styled Component 를 사용하여 CSS 를 적용시켰으며, Hover 시 크기가 커지는 애니메이션을 적용하였습니다.

페이지 이동방식은 라우팅 방식을 사용하여 navigate() 함수를 통해 이동하도록 하였습니다.

모든 페이지는 반응형 웹으로 구현하였습니다.

## 3-2. Dictionary Page



[Dictionary Page]

### 3-2-1. 페이지 설명

Dictionary Page 는 물고기들의 정보를 확인할 수 있는 페이지입니다. 물고기들을 카드 형태로 배치하여 사용자가 다양한 물고기들을 쉽게 둘러볼 수 있도록 하였습니다. 카드를 클릭하면 해당 물고기의 정보를 알려주는 페이지로 이동하여 물고기 정보를 확인할 수 있도록 하였습니다.

검색 기능을 구현하여 물고기의 이름을 입력하여 해당 물고기를 찾을 수 있도록 하였으며, 물고기의 이름 전체를 입력하지 않아도 해당 글자가 포함된 물고기가 표시되도록 하였습니다.

페이지 상단에는 Header 를 배치하여 Aqua Mate 로고와 현재 페이지 아이콘, 내비게이션 버튼을 표시하였습니다. 좌측 상단의 Aqua Mate 로고를 클릭하여 User Home Page 로 이동할 수 있으며, 중앙의

현재 페이지 아이콘을 클릭하여 페이지를 새로고침할 수 있습니다. 우측 상단의 내비게이션 버튼을 클릭하여 다른 페이지로 이동할 수 있습니다.

### 3-2-2. 기술 설명

---

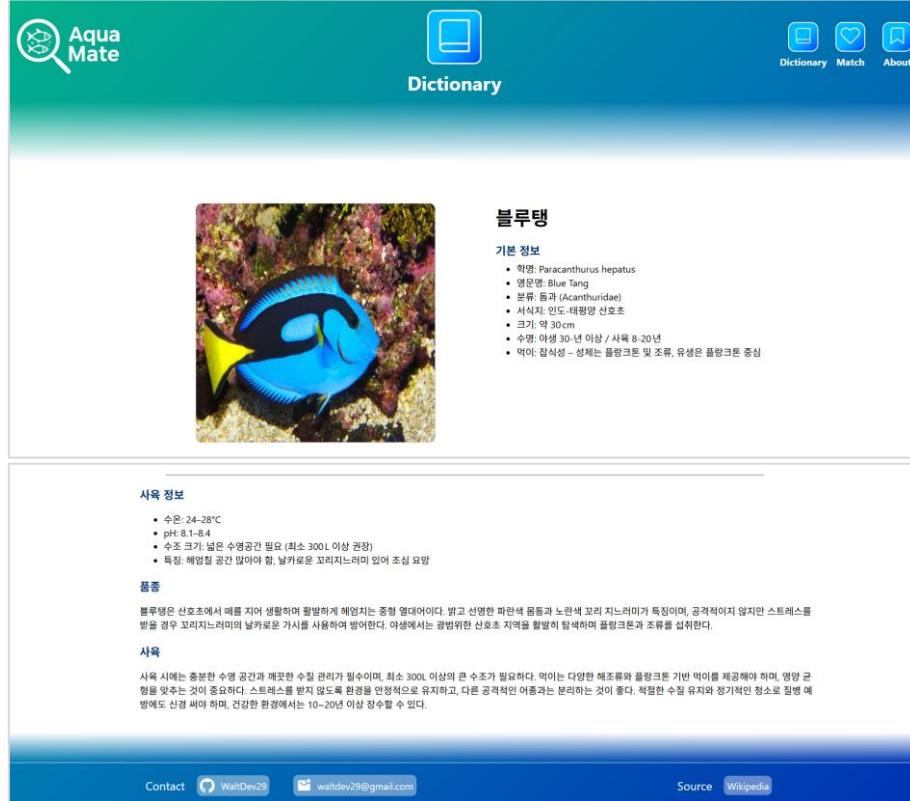
본 페이지에도 Layout.jsx 컴포넌트를 적용하였으며, Header 가 렌더링 되도록 설정하였습니다. 추가적으로 화면 중앙의 콘텐츠 배경과 자연스럽게 연결되도록 Gradient.jsx 컴포넌트를 만들어 그라데이션을 추가하였으며, Layout.jsx 와 같이 컴포넌트 합성을 활용하였습니다. User Home Page 에서 사용하였던 NavBtn.jsx 컴포넌트를 재사용하였으며, props 로 className 을 전달하여 다른 CSS 가 적용되도록 하였습니다.

검색 영역은 컴포넌트로 분리하였으며, 상위 컴포넌트인 DictionaryPage.jsx 에서 props 로 setState 함수들을 전달하여 DictionaryPage.jsx 의 state 를 변경할 수 있도록 하였습니다.

물고기 데이터는 json 파일로 저장하여 DictionaryPage.jsx 컴포넌트에서 Fetch() 함수를 통해 불러왔으며, 사용자가 검색창에 입력한 문자를 포함한 데이터 Filter() 함수로 모아 state 로 관리하였습니다. 해당 state 를 map() 함수를 통해 물고기 카드로 렌더링 하였으며, 물고기 카드는 FishCard.jsx 컴포넌트로 구현하였습니다.

물고기 카드 클릭 시 navigate() 함수를 통해 물고기 정보 페이지로 이동하며, 해당 물고기 정보를 페이지에 전달하여 표시하도록 하였습니다. 물고기 카드의 즐겨찾기 버튼은 기능을 구현하지는 않았지만, 클릭 시 버튼의 색이 바뀌도록 하였습니다. 물고기 카드 Hover 시 크기가 커지는 애니메이션을 설정하였습니다.

### 3-3. Fish Info Page



[Fish Info Page]

#### 3-3-1. 페이지 설명

Fish Info Page 는 물고기의 정보를 보여주는 페이지입니다. 사용자가 Dictionary Page, Match 페이지에서 클릭한 물고기의 정보를 제공합니다.

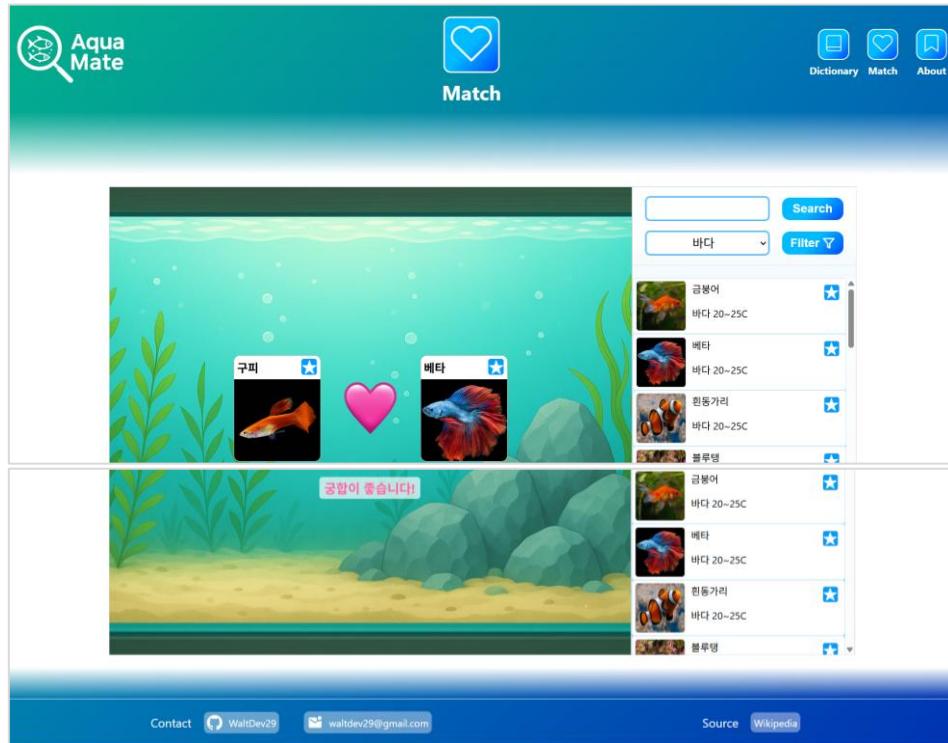
#### 3-3-2. 기술 설명

useLocation() Hook 을 사용하여 이전 페이지에서 전달한 물고기 정보를 받아 해당 물고기 정보를 표시하도록 하였습니다.

물고기 정보 중 리스트 형식의 정보들은 map() 함수를 사용하여 렌더링 하였습니다.

## 3-4. Match Page

---



[Match Page]

### 3-4-1. 페이지 설명

---

Match Page는 물고기들의 궁합을 확인할 수 있는 페이지입니다. 물고기 목록에서 물고기 두 마리를 선택하여 궁합을 확인할 수 있습니다. (궁합 분석 기능 미구현)

물고기 목록에서 물고기를 검색할 수 있고, 물고기 목록에서 물고기 이름을 선택하거나, 선택된 물고기 카드를 클릭하여 해당 물고기의 정보 페이지로 이동할 수 있습니다.

### 3-4-2. 기술 설명

---

Aside.jsx 컴포넌트를 만들어, Dictionary Page 와 같은 방식으로 검색 기능을 구현하고 물고기 목록을 렌더링 하였습니다. 물고기 목록 Hover 시 색상을 어둡게 하여 선택할 수 있음을 나타냈습니다.

Match Page 에 state 를 두고 하위 컴포넌트인 Aside.jsx 컴포넌트에 setState 함수를 전달하여, 물고기 목록 클릭 시 해당 물고기가 Match Page 의 state 로 설정되도록 하였습니다. 이 state 는 배열 형태로 사용하여 두 마리의 물고기 정보를 담도록 하였으며, 이미 두 마리가 선택된 상태에서는 새로운 물고기 선택 시 배열이 초기화되도록 하였습니다.

### 3-5. About Page



[About Page]

### **3-5-1. 페이지 설명**

---

About Page 는 Aqua Mate 를 소개하는 페이지입니다. Aqua Mate 의 취지와 기능들을 간단하게 소개합니다.

### **3-5-2. 기술 설명**

---

About Page 에는 내비게이션 메뉴를 클릭하면 해당 소개 스크롤로 이동하도록 하였습니다.

## 4. 고찰

저는 코드의 중복을 줄이는 것을 좋아합니다. 특히 반복되는 코드를 함수로 만들고 매개변수를 설정할 때면 즐거움을 느낍니다. 하지만 HTML은 반복되는 태그를 하나로 통합할 수 없다는 점이 아쉬웠고, JavaScript로 태그를 동적으로 생성할 때도 번거롭고 직관적이지 못하다고 느꼈습니다. 그래서 2학기에 처음 리액트를 배우기 시작하였을 때 HTML 태그들을 함수에 담아 재사용하는 방식을 접하여 매우 반가움을 느꼈습니다.

이번 기말고사 대체 과제를 수행하면서도 1학기에 만들었던 페이지를 보면 리액트의 편리함을 다시금 실감하였습니다. 기존에 제작한 웹사이트는 하나의 파일에 페이지의 모든 태그들이 담겨 있었고, 그에 대응하는 css 파일과 js 파일이 하나씩 있는 구조였습니다. 따라서 페이지의 내용이 많아질수록 HTML 태그의 구조가 복잡해졌으며, CSS와 JavaScript 적용을 위해 여러 태그의 ID와 Class 명을 직관적이고 간결하며 겹치지 않도록 신경 써야 했습니다. 하지만 리액트에서는 여러 태그를 하나의 컴포넌트로 묶어 관리할 수 있었으며, Styled Component를 사용하여 하나의 jsx 파일 안에 HTML 태그, CSS, JavaScript를 모두 담을 수 있는 것이 편리하였습니다.

웹 UI를 구성하는 작업은 저에게는 늘 번거롭게 느껴졌는데, 이번 학기 리액트 수업을 통해 빠르고 효율적으로 UI를 만드는 방법을 배울 수 있었습니다. 한 학기 동안 귀한 강의를 해주셔서 감사드리며, 내년에도 성실히 임하겠습니다. 감사합니다.