

이미지 캐러셀(Carousel) 만들기

1. 과제 개요

과제 평가 단계에서 제출하셔야 할 산출물은 **이미지 캐러셀 제작**입니다. 본 과제는 **제출 기한을 준수하여, 요구사항을 충족하는 애플리케이션을 제작하는** 것을 목표로 합니다.

이미지 캐러셀(Carousel)은 여러 장의 이미지를 일정한 영역 내에서 순차적으로 전환하며 표시하는 UI 컴포넌트로, 사용자가 좌우 버튼이나 스와이프 등을 통해 직접 이미지를 넘길 수 있으며, 자동 재생(Auto Slide) 기능을 통해 연속적인 슬라이드쇼를 제공할 수 있습니다. 웹 페이지의 시각적 집중도를 높이고 제한된 공간에서 다양한 콘텐츠를 효율적으로 제공하기 위해 널리 활용되는 인터페이스 요소로, 전환 애니메이션, 네비게이션 버튼, 페이징 인디케이터 등 다양한 조작 요소가 포함될 수 있습니다.

요구사항 외에도 **개발 환경, 유의사항, 제출 방법**을 함께 확인해주시기 바랍니다. 또한 **평가 기준**이 첨부되어 있으니, 과제 수행 시 참고해주시기 바랍니다.

2. 제출 방법

과제 수행 기간은 **2025.12.13 10:00 AM - 2025.12.14 10:00 AM**입니다.

아래 절차에 따라 과제 코드를 제출해주시기 바랍니다.

- 수행한 코드를 본인 GitHub 저장소에 **Private Repository**로 생성한 뒤 push합니다.
- 해당 저장소의 **Collaborator**로 **telepixcompany** 계정을 추가합니다.
- 제출 기간 내에 제공된 **Google Form**에 GitHub 저장소 주소를 기입하여 제출합니다.

3. 요구 사항 및 유의 사항

기본 요구 사항

- 해결 전략**을 README.md에 명확히 기술합니다.
- 모든 코드는 **TypeScript**로 작성합니다.
- 웹 접근성 기준**을 준수하여 애플리케이션을 구현합니다.
- 현재 제공되는 라이브러리 외의 라이브러리를 사용 가능합니다.
 - 단, 과제의 **구현 목표와 직접적으로 연관된** 라이브러리는 사용할 수 없습니다.

- 현재 제공된 라이브러리 외의 라이브러리를 설치하는 경우, **해결전략** (**README.md**)에 라이브러리를 채택한 이유를 작성해주세요.
-

과제 요구 사항

- 아래에 제시된 핵심 기능을 모두 개발합니다.
 - **기본 기능**
 - 여러 장의 이미지를 **API 요청을 통해 불러와 순차적으로 표시하는 이미지 캐러셀 구현**
 - 마우스 클릭을 통한 **이전(◀) / 다음(▶)** 이미지 이동 기능
 - 키보드 방향키(**← / →**)를 통한 이미지 이동 기능
 - 현재 이미지 위치를 나타내는 **페이지 인디케이터(●)** 표시 기능
 - 이미지 전환 시 적용되는 **부드러운 전환 애니메이션(transition)** 제공 기능
 - **자동 재생 기능 (Auto Slide)**
 - 일정 시간 간격(예: 3초)으로 다음 이미지로 이동하는 **자동 재생** 기능
 - 자동 재생 중 Auto 버튼 클릭 시 자동 재생을 일시정지하는 **Pause** 기능
 - 일시정지 이후 자동 재생을 다시 시작하는 **재시작** 기능
-

유의 사항

- 현재 제공되는 라이브러리 외의 라이브러리를 사용 가능합니다.
 - 단, 과제의 구현 목표와 직접적으로 연관된 라이브러리는 사용할 수 없습니다.
 - **swiper**, **react-slick**, **embla-carousel**, **keen-slider** 등 외부 캐러셀·애니메이션 라이브러리 관련 외부 사용 금지
 - 현재 제공된 라이브러리 외의 라이브러리를 설치하는 경우, **해결전략** (**README.md**)에 라이브러리를 채택한 이유를 작성해주세요.
- UI 디자인에는 제한이 없으나, **사용자 경험(UX)**을 충분히 고려하여 설계해주시기 바랍니다.
- 실제 업무 환경에서 개발한다는 관점으로 **실무 수준의 품질**을 목표로 수행해주시기 바랍니다.
 - 예시 : lint, prettier, git commit 작성 등
- API 연동은 **3. 기술 및 환경 설정** 항목을 확인한 뒤 진행해주시기 바랍니다.

- 모든 요구사항을 완벽히 충족하지 않더라도 제출이 가능합니다.
- AI Agent 사용을 금지합니다.
- 과제 제출 시 GitHub 저장소에 **Private Repository**가 아닌 **Public Repository**로 생성하였을 경우 불이익이 있을 수 있습니다.
- 과제 평가는 제출 기간 내에 Commit한 내용만 반영되며, 이후 작업은 반영되지 않습니다.

4. 기술 및 환경 설정

기본 URL: <http://localhost:3002>

기본 설정: json-server 설정

1. json-server 설치

```
npm install -g json-server
```

2. json-server 실행

```
//package.json
"server": "json-server --watch db.json --port 3002"
```

API 엔드포인트

1. 이미지 조회

a. URL: <GET /images>

b. Response

```
[
  {
    id: string;
    author: string;
    url: string;
```

```
    }  
]
```

5. 개발 환경

개발 환경은 아래와 같습니다. 보다 상세한 환경 구성은 제공된 템플릿 코드의 package.json 파일을 참고해주시기 바랍니다.

언어	TypeScript v5.9
기본 라이브러리	Next.js v15.5 (app router)
전역상태 라이브러리	<code>zustand v5</code> (전역상태 라이브러리를 사용하는 경우에 사용, 전역 상태 라이브러리 사용 여부는 선택사항)
스타일 라이브러리	자유롭게 선택 가능
테스트 환경	Chrome 142.0.7444.176(arm64)

6. 평가 기준

본 과제의 평가는 기능 구현 수준뿐만 아니라 코드 품질, 유지보수 가능성, 사용자 경험 설계 등 전반적인 개발 역량을 종합적으로 확인하는 것을 목표로 합니다.

1) 기능 구현 및 완성도

- 요구사항을 충족하여 애플리케이션을 구현하였는지 평가합니다.
- 애플리케이션의 전반적인 완성도를 평가합니다.
(예: 여러 핸들링 적절성, 기능의 안정적 동작 여부 등)

2) 가독성, 유지보수성 및 확장성

- 코드의 가독성을 평가합니다.
- 향후 유지보수 및 기능 확장을 고려하여 구조화되어 있는지를 평가합니다.

3) UX를 고려한 디자인 설계

- UI 구성 방식에는 제한이 없습니다.
- 단, 사용자 편의성과 흐름을 고려한 **UX 중심의 설계**가 이루어졌는지 평가합니다.

- 접근성을 고려한 디자인 설계를 평가합니다.