리액트 프로젝트 보고서

# 1) 프로젝트 명칭, 프로젝트 작성자

**프로젝트 명칭: LIfeControlApp**

**작성자: 2학년 YB반 20220876심민식**

# 2) 주제 선정 이유 (개발 필요성)

최근 교내 도서관에서 시행한 독후감 공모전을 계기로 ‘프리드히 니체’의 위버멘쉬라는 책을 읽게 되었습니다. 위버멘쉬란 니체가 궁극적으로 목표로 하는 인간상인데, 끊임없는 성장을 강조하는 인간상이었습니다. 그 책을 감명깊게 읽고 조금씩이라도 성장을 하려면 무엇이 필요할까 고뇌하던 중 본인이 읽거나 시청한 영화를 보고 생각을 정리할 수 있도록 후기를 작성하는 페이지를 만들고, 운동을 기록하는 페이지, 일정을 관리할 수 있도록 캘린더에 일정을 삽입하는 페이지, 하루를 되돌아보며 성찰할 수 있도록 일기페이지를 만들어 생활을 컨트롤 하고 더욱 성장할 수 있는 앱을 만들었습니다.

# 3) 개발 환경 (프로젝트 개발 환경 명시)

-프런트엔드 프레임워크: 리액트

-개발 서버: vite

-언어: html, JavaScript

-스타일링: Tailwind Css V4

-패키지 매니저: npm 10

-에디터: Visual Studio Code

-운영 체제: macOS

-브라우저: Chrome

-다운로드한 라이브러리: tailWindCss postcss autoprefixer, dayjs, react-router

# 4) 각 모듈 개략적 설명 (프로젝트 어떻게 구현되었는지 간략 설명)

페이지 구성: 메인페이지, 책 리뷰 페이지, 영화 리뷰 페이지, 캘린더 페이지. 일기 페이지

페이지 공통 사항: 각 페이지로 이동할 수 있는 네비게이션 바 삽입, TailWindcss 를 사용하여 넷플릭스의 색감을 참고하여 디자인을 구성했습니다.

-메인페이지

책 리뷰 페이지 / 영화 리뷰 페이지 / 운동기록 페이지 / 캘린더 페이지 / 일기 페이지로 이동할 수 있는 페이지이고 디자인은 넷플릭스의 계정 선택 페이지를 참고 했습니다.

-책 / 영화 리뷰 페이지: 조건부 렌더링을 통하여 리뷰를 작성하는 페이지를 만들었습니다. 책 부분에서는 책의 리뷰를 작성할 수 있도록 하였고, 영화 부분에서는 영화의 리뷰를 작성할 수 있도록 구성하였습니다.

-운동기록 페이지: 날짜를 선택하고, 수행한 운동을 선택하여 추가할 수 있는 페이지를 만들었습니다. 일정 추가 하단에는 그래프를 만들어 일주일내에 운동한 양을 볼 수 있도록 하였습니다.

-캘린더 페이지: 달력에서 일정을 추가하고 삭제 할 수 있도록 표현하였습니다.

-일기 페이지: 일기 추가를 누르면 일기를 작성할 수 있게 구현하였습니다.

# 5) 추후 활용 계획 또는 해당 프로젝트 향후 보완할 사항

Json을 사용하여 데이터를 저장하고 불러오는 방법을 사용하지 못 하였습니다.

Json을 통해 데이터 저장이 용이하도록 변경할 계획입니다.

그렇게 정보를 저장할 수 있게 만들어 앞으로의 일상을 기록할 생각입니다.

그리고 이미지가 하나도 들어가지 않아서 이미지를 사용해 시각적으로 조금 더 보기 좋게 만들 계획입니다.

# 6) 프로젝트 실행 방법: 사용 매뉴얼

npm i

npm install -D tailwindcss postcss autoprefixer

npm install dayjs

npm install react-router-dom

npm run dev

# 7) 프로젝트 실행 화면 캡처: 사용 매뉴얼

1. 메인페이지

-처음 화면: 각각의 카테고리를 선택

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

-각각의 주제에 마우스를 가져다 대었을 경우

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

2. 책 / 영화 리뷰 페이지

-처음 페이지에 들어갔을 경우

책 리뷰 화면

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

영화 리뷰 화면

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

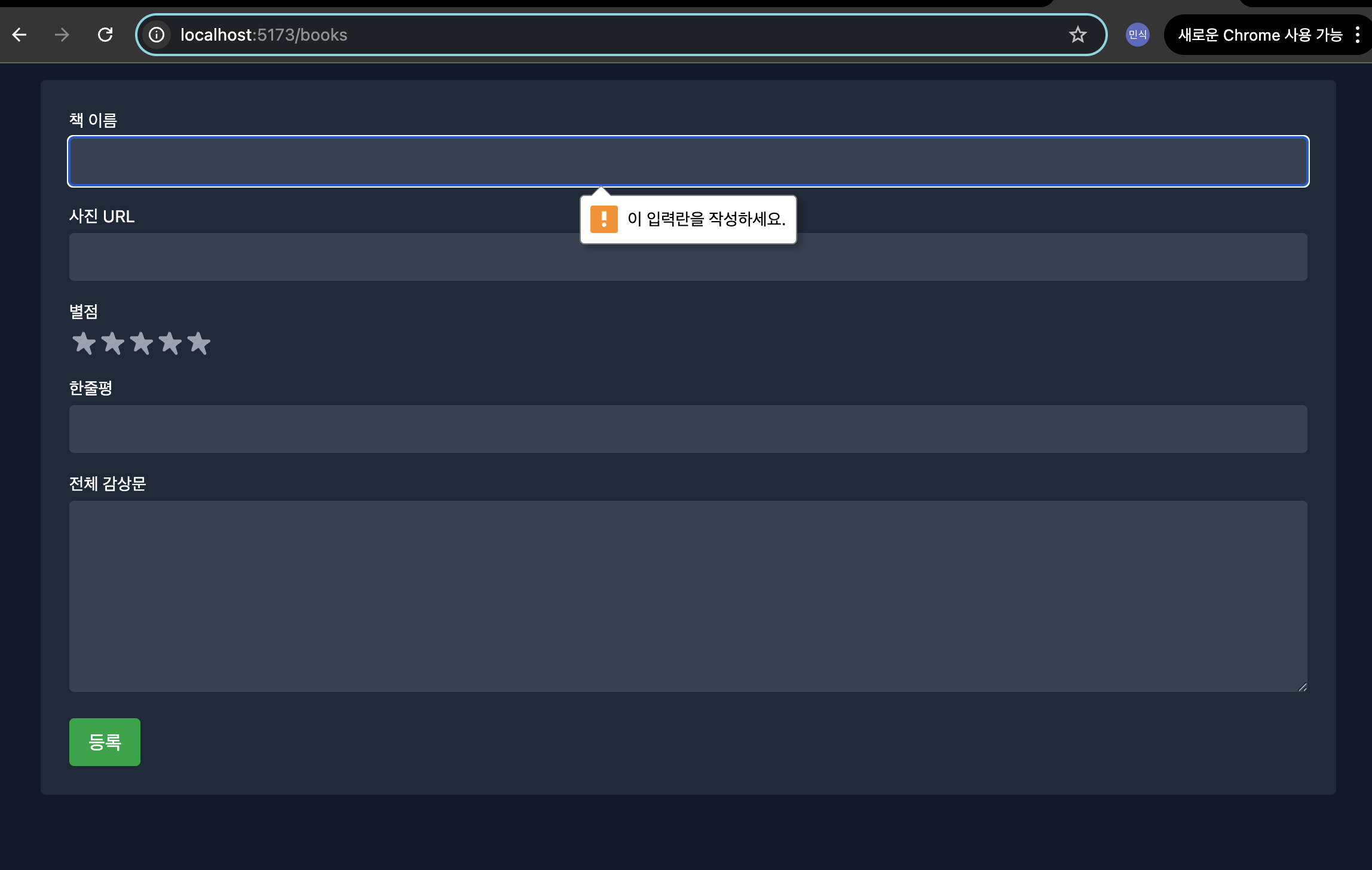
AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

-리뷰 추가를 눌렀을 경우

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

-아무것도 입력하지 않고 등록을 눌렀을 경우

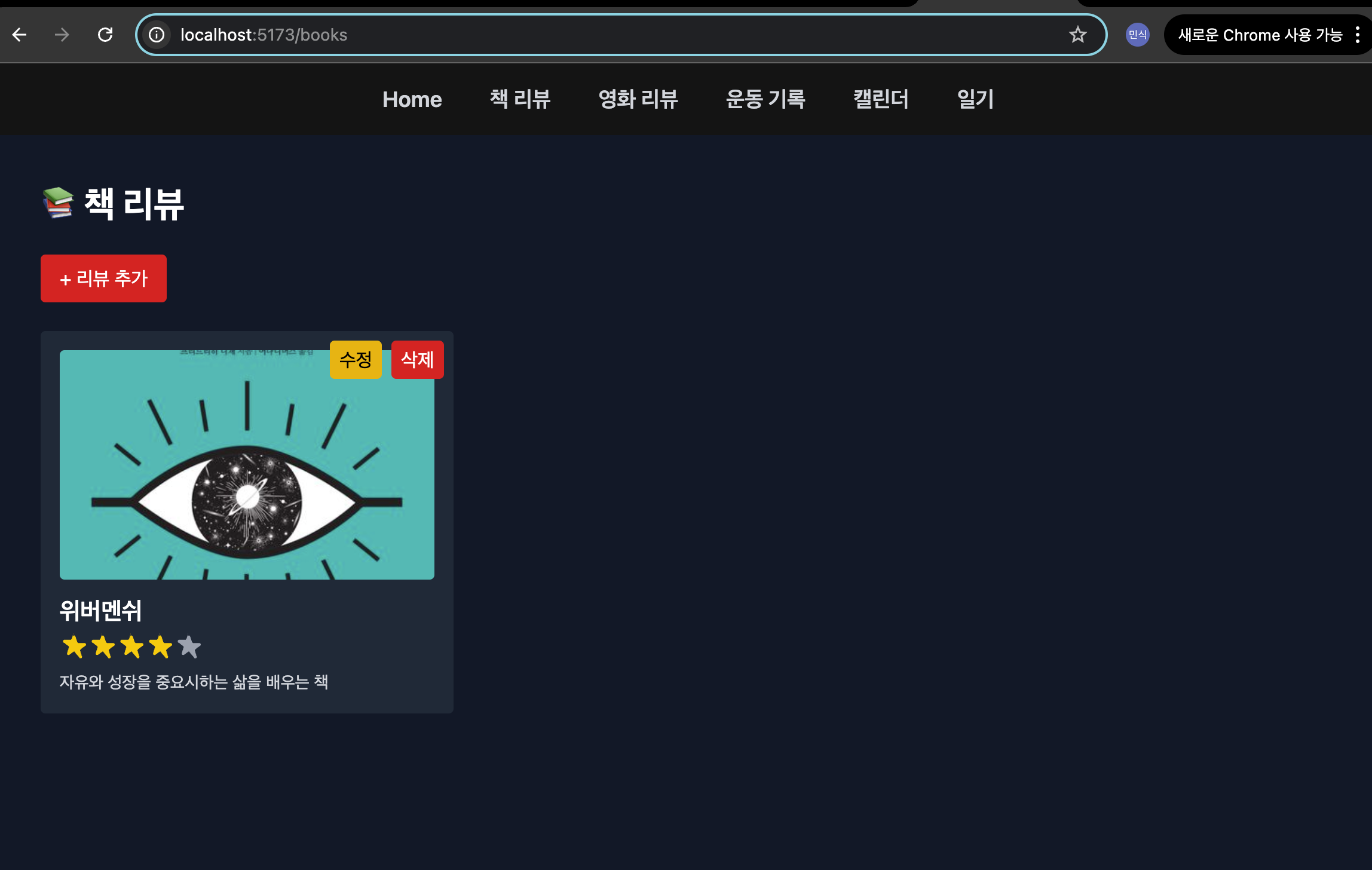


-리뷰 내용 작성

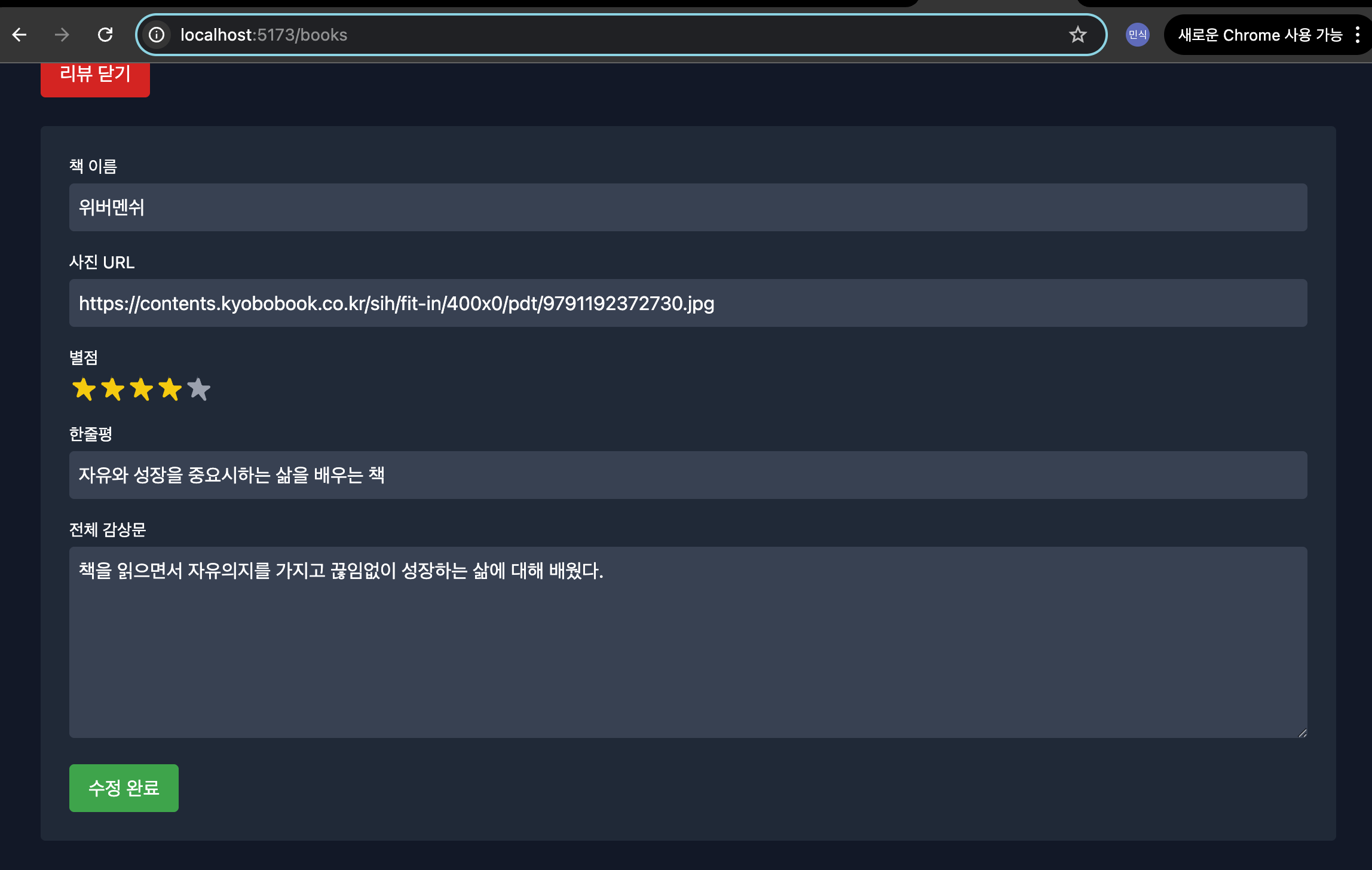
텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

-리뷰 작성 후



-수정버튼 클릭 시

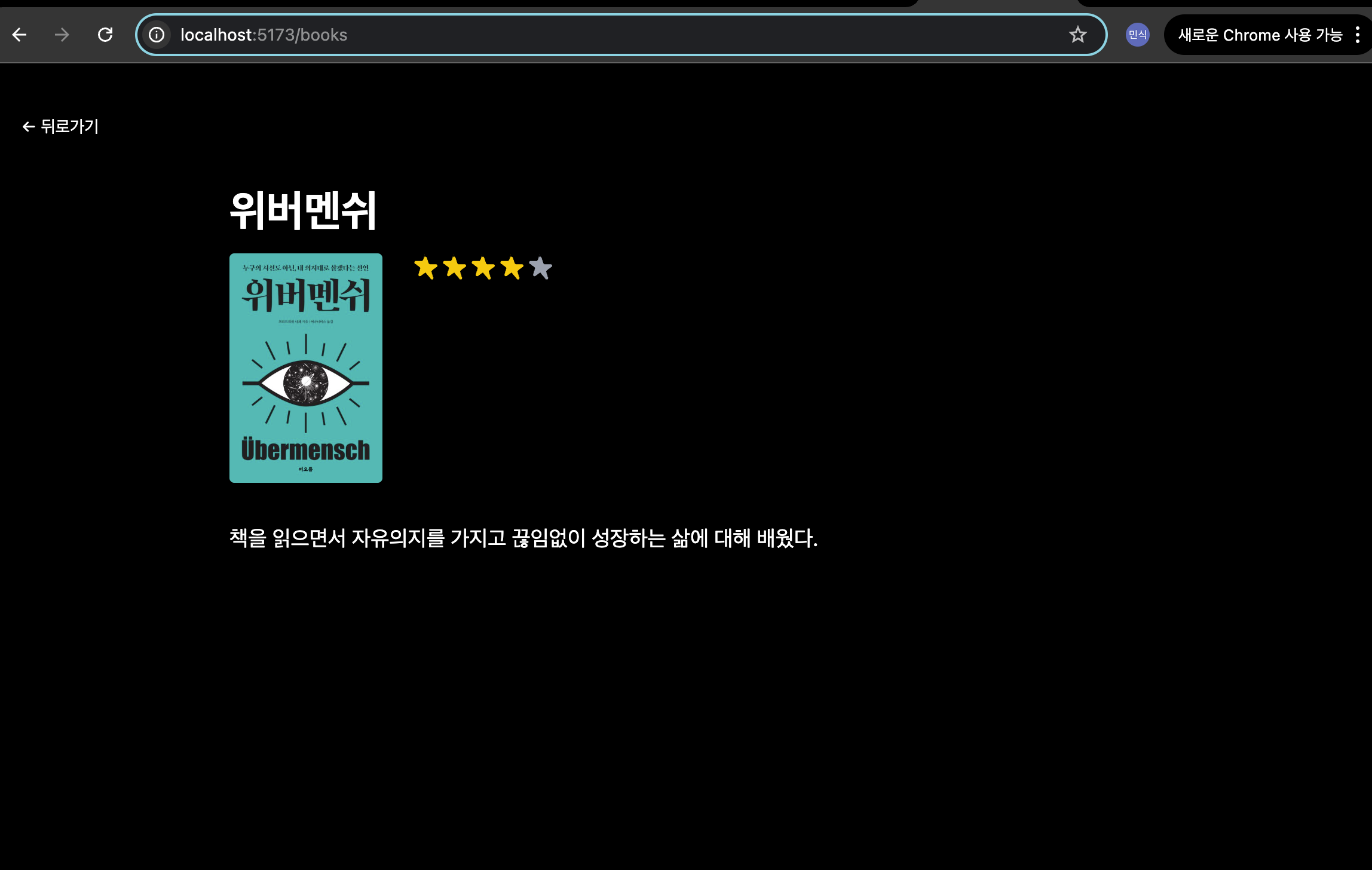


-삭제버튼 클릭 시

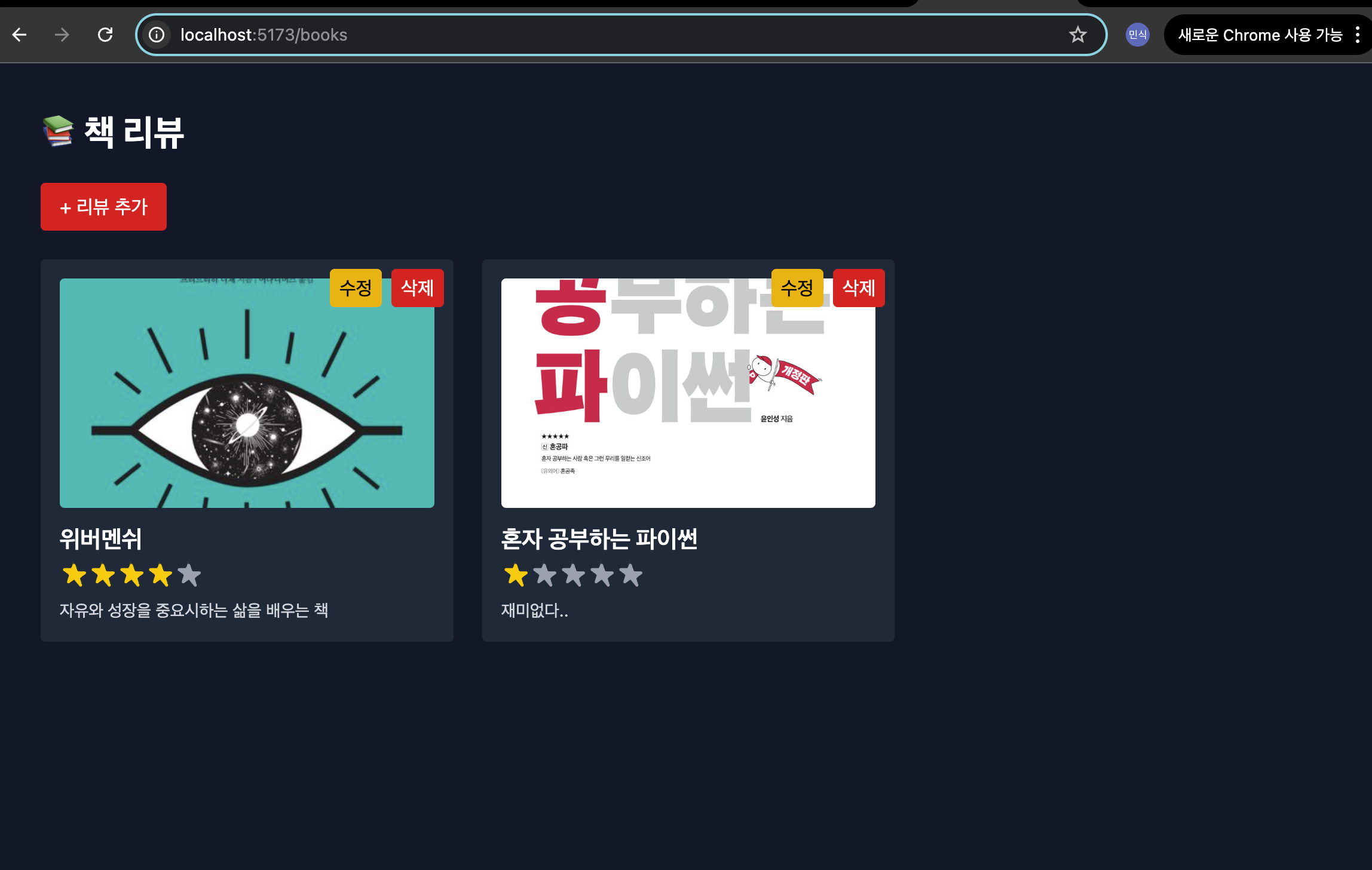
텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

-작성한 리뷰를 선택 시

****

**-리뷰가 여러개일 경우 별점순으로 정렬**

****

**3. 운동기록 페이지**

**-처음 페이지 접속 화면**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-날짜 선택**

**소프트웨어, 텍스트, 멀티미디어 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-운동 선택**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-추가 후 화면**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-직접 입력 선택 화면**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-후기 여러개 작성 후 화면**

**스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-수정버튼 클릭 후 화면**

**스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**4. 캘린더 페이지**

**-처음 페이지 접속 화면**

**스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-날짜 선택 화면**

**스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-카테고리 선택화면**

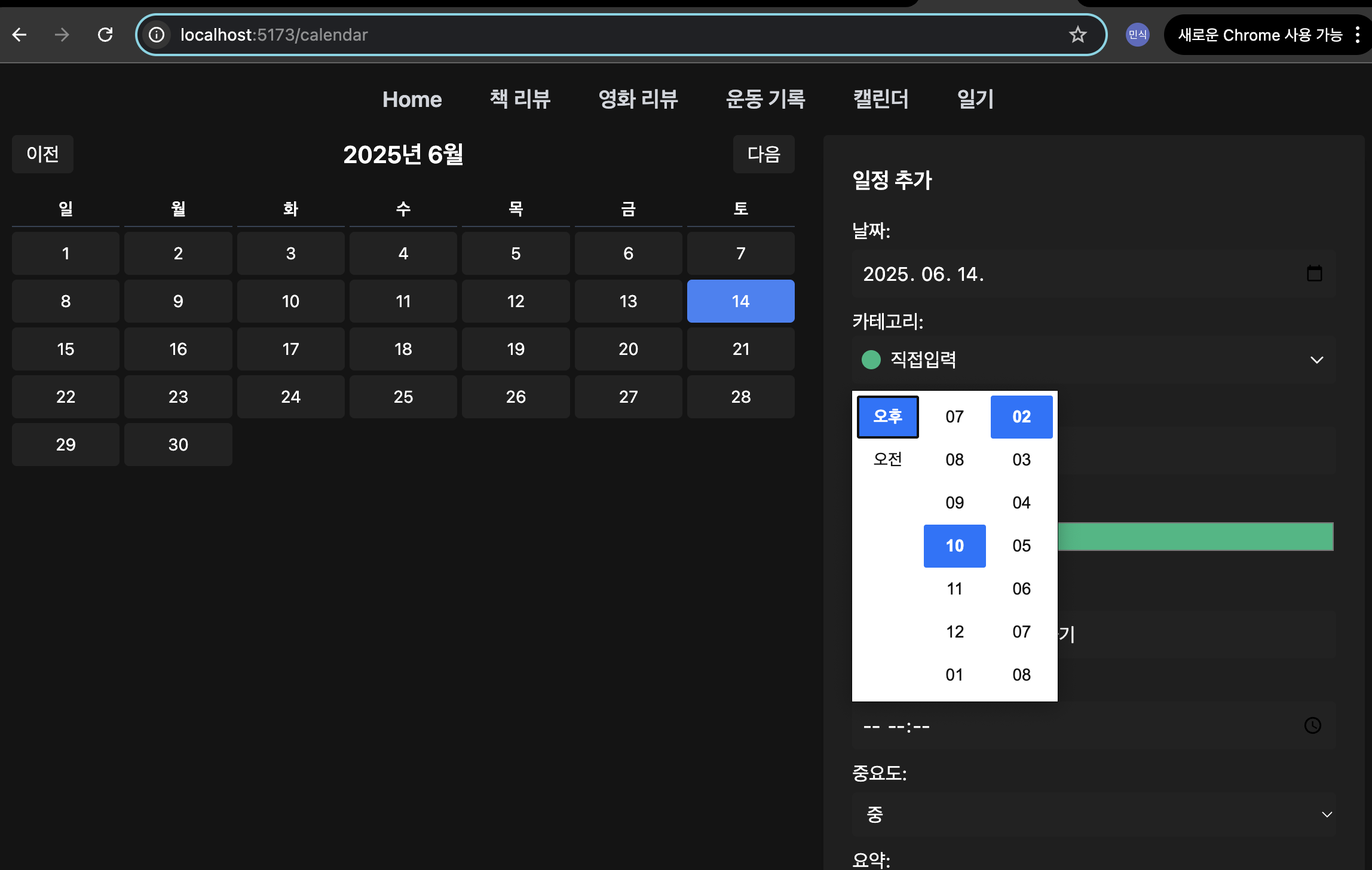
**스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-직접입력 선택 시 화면**

****

**-시간 선택 화면**

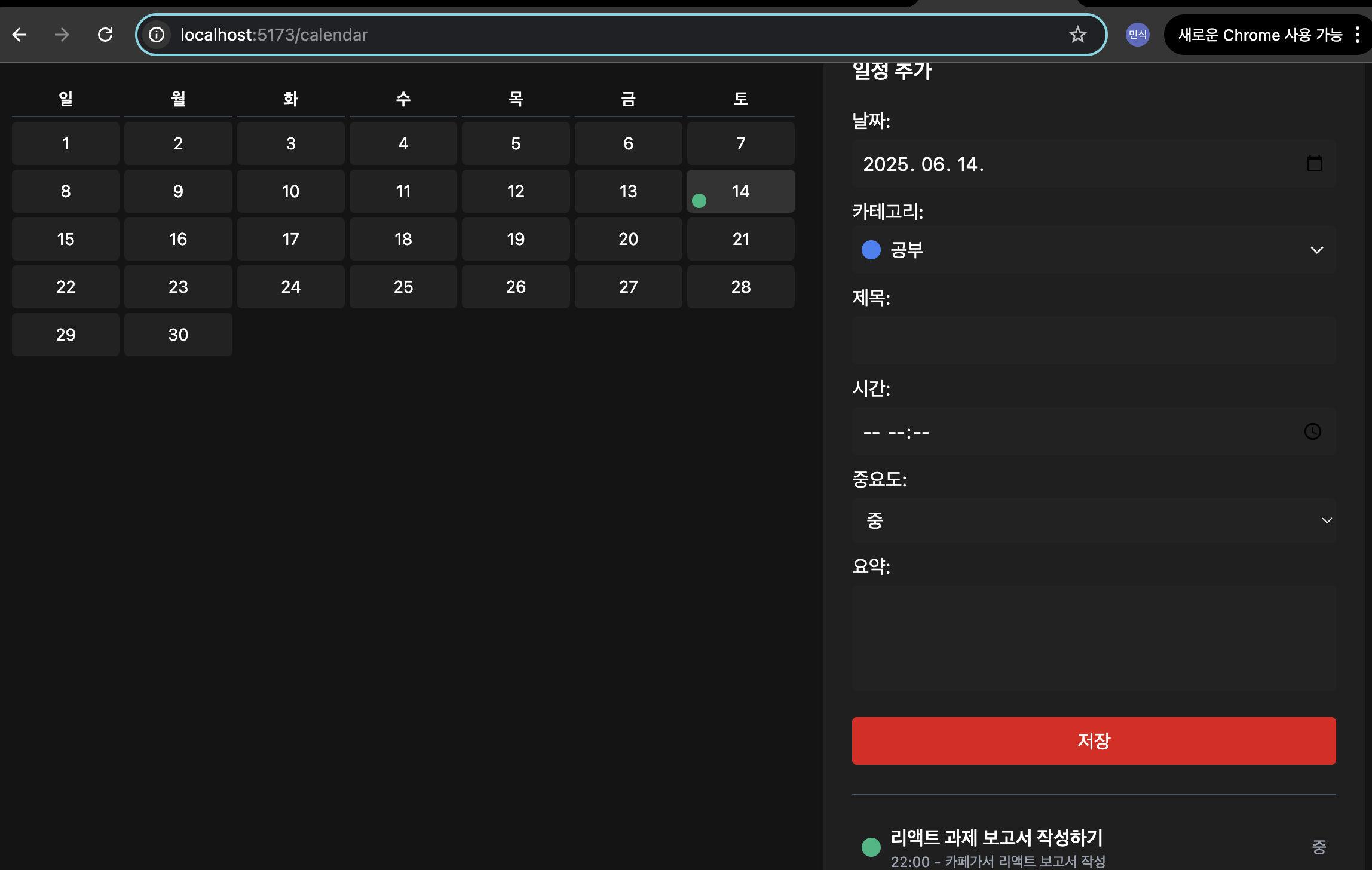
****

**-중요도 선택 화면**

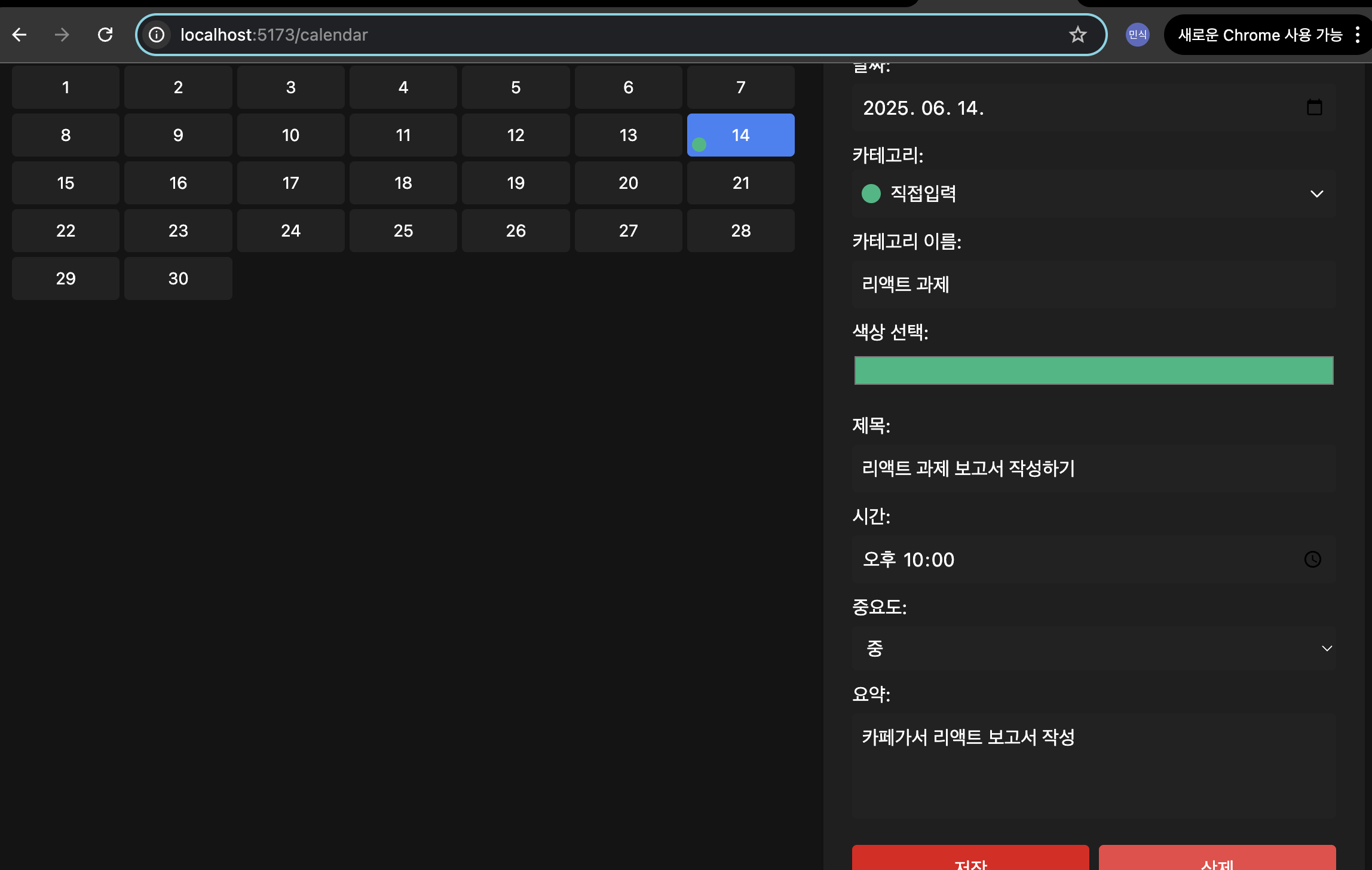
**스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-일정 추가 후 화면**

****

**-작성된 일정 클릭 후 화면**

****

**5.일기 페이지**

**-페이지 접속 후 화면**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-일기 추가 클릭 후 화면**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-일기 추가 후 화면**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

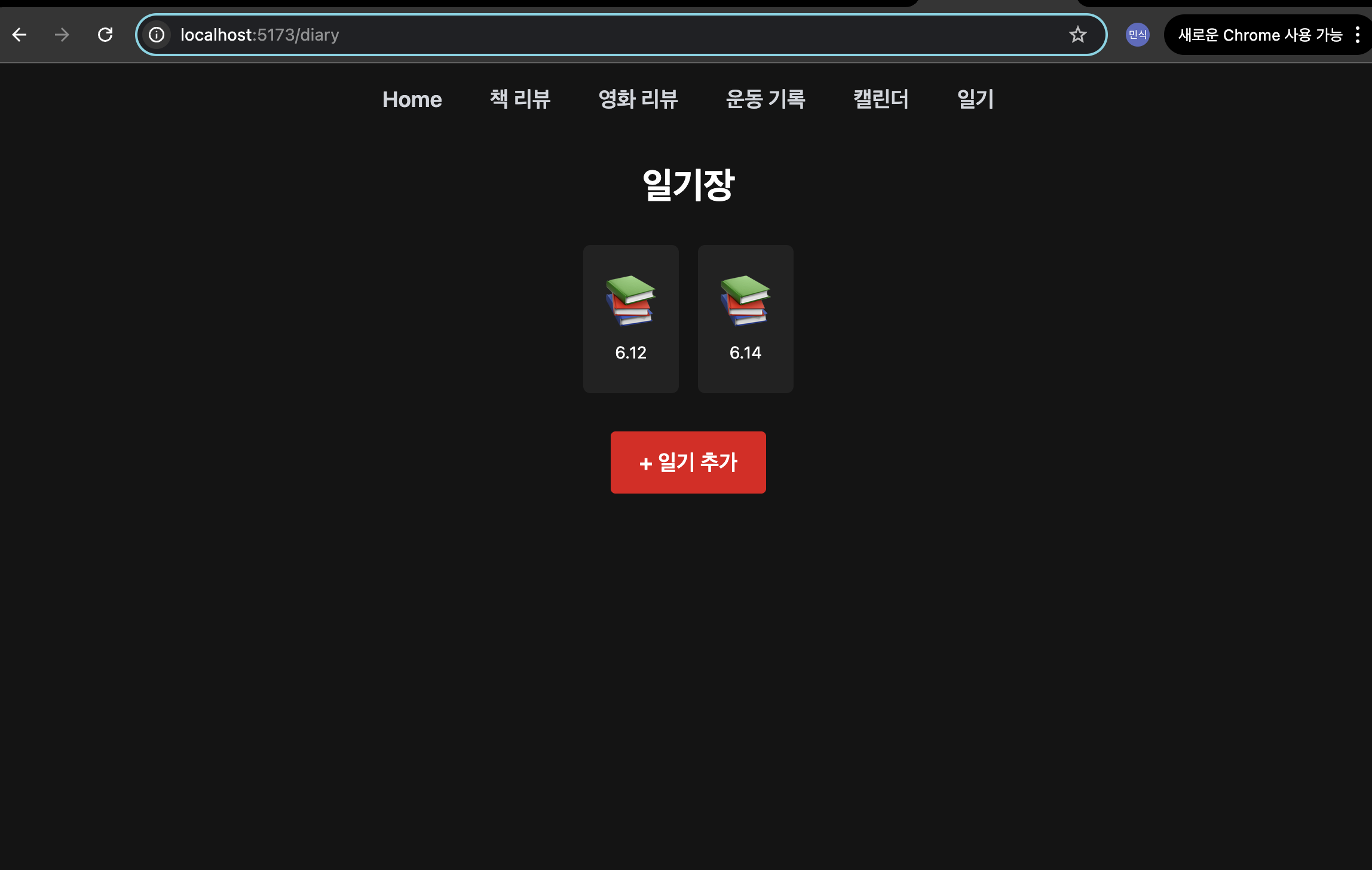
AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-작성된 일기 클릭 후 화면**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**-일기 여러개 작성 후 화면(날짜 순으로 정렬)**

****

**-삭제 버튼 누른 후 화면**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

# 8) 구현한 기술 구체적으로 설명하는 내용 작성

공통

-리액트 라우터를 사용하여 페이지 간 렌더링 / Link to를 사용하여 네비바를 헤더로 구성

1.리뷰 페이지

- useState를 통해 상태값 선언

reviews: 리뷰 객체들의 배열

setReviews: 리뷰 배열을 갱신하기 위한 setter함수

title: 리뷰할 책/영화의 제목 문자열

image: 리뷰할 책/영화의 이미지 url

rating: 1~5사이의 정수형 별점 값

shortReview: 상세 감상문

editIndex: 현재 수정 중인 리뷰의 배열 인덱스, 수정 중이 아니면 null

-리뷰 추가/수정

if (!title || !rating || !shortReview || !longReview) return; : !연산자를 통해 각 항목이 비어있으면 true, 후 리턴

const newReview = { title, image, rating, shortReview, longReview }; : 리뷰 객체 생성

if (editIndex !== null): editIndex가 수정모드인지 확인, spread(…reviews)로 기존 배열 복사, 기존인덱스에 새 리뷰 삽입, 아니면 새로운 배열로 리뷰 추가

-리뷰 수정

handleEditReview: 클릭된 인덱스의 리뷰 데이터를 각각 상태값에 설정, setEditIndex로 수정 모드 진입

-리뷰 삭제

HandleDeleteReview: filter()메서드를 사용해 인덱스가 일치하지 않는 항목만 남김, i는 현재 인덱스를 의미, 수정모드 중이면 editIndex초기화

-ui 구성요소

1. 입력 폼

2. 리뷰 목록 렌더링

-map()을 사용하여 배열을 리스트로 변환

-수정/삭제 버튼에 index 전달하여 정확한 항목 식별

3.감상문 상세 보기

-선택된 리뷰를 따로 selectedReview 상태로 관리, 클릭시 감상 문 및 이미지 출력

- 조건부 렌더링

showForm === true → 리뷰 작성 폼 렌더링

selectedReview !== null → 상세 리뷰 보기 렌더링

reviews.length > 0 → 리뷰 카드 리스트 렌더링

2.운동 페이지

-useState를 통해 상태값 선언

workouts: 사용자의 운동 기록 목록을 저장하는 배열 상태

input: 운동 선택 드롭다운의 현재 값 (기본 운동 또는 '직접 입력')

date: 사용자가 선택한 날짜

customInput: 사용자가 '직접 입력'을 선택했을 때 입력하는 운동 이름

editIndex: 현재 수정 중인 운동 항목의 인덱스 (null이면 새 항목 추가 상태)

-미리 정의된 운동 항목 목록

predefined: ['데드리프트', '벤치프레스', '스쿼트', '러닝', '직접 입력']

-운동 추가/수정

handleAdd:

- input이 '직접 입력'일 경우 customInput 사용, 그렇지 않으면 input 값 사용

- activity 또는 date가 비어 있으면 early return

- 새 운동 기록 객체 생성: { date, activity }

- 수정 모드(editIndex가 null이 아님)면 해당 인덱스의 항목을 업데이트

- 아니라면 workouts 배열에 새 기록 추가

- 작업 후 date, input, customInput, editIndex 모두 초기화

-운동 수정

handleEdit(index):

- 선택한 인덱스의 운동 기록을 불러옴

- predefined 목록에 포함되어 있으면 그대로 input에 설정

- 포함되어 있지 않으면 input을 '직접 입력'으로 설정하고 customInput에 값 입력

- editIndex를 해당 인덱스로 설정하여 수정 모드 진입

-운동 삭제

handleDelete(index):

- workouts에서 선택한 인덱스를 제거한 새 배열 생성

- setWorkouts로 상태 갱신

- editIndex를 null로 초기화

-최근 7일 날짜 배열 생성 함수

getLast7Days():

- 오늘부터 6일 전까지 날짜를 for문으로 반복

- 각 날짜를 ISO 문자열(YYYY-MM-DD)로 변환하여 배열에 저장

- 최종 결과: ['2025-06-08', '2025-06-09', ..., '2025-06-14']

-최근 7일 요약 데이터

summary:

- last7Days 배열을 기반으로 reduce 실행

- 각 날짜에 해당하는 운동 기록 수를 계산

- { '2025-06-13': 2, '2025-06-14': 0, ... } 형태의 객체 반환

-UI 구성 요소

1. 입력 폼:

- 날짜 입력 input

- 운동 선택 select 드롭다운 (predefined 목록 사용)

- '직접 입력' 선택 시 텍스트 필드 추가 렌더링

- '추가' 버튼 클릭 시 handleAdd 실행

- 수정 중이면 버튼 텍스트가 '수정'으로 변경됨

2. 운동 기록 리스트:

- workouts 배열을 순회하며 날짜 + 운동명 표시

- 각 항목에 '수정' / '삭제' 버튼 포함

- '수정' 클릭 시 handleEdit 실행 → 폼에 반영

- '삭제' 클릭 시 handleDelete 실행

3. 최근 7일 운동 그래프:

- last7Days 배열 순회

- 각 날짜별로 요약 데이터(summary[d])에 따라 막대 길이 설정

- 1회당 25px로 width 계산 (style={{ width: `${summary[d] \* 25}px` }})

3. 일기 페이지

-useState를 통한 상태값 선언

diarides: key가 날짜, value가 내용인 객체, ustState에 콜백함수를 넣어 초기화 시 컴포넌트 첫 렌더링 시에만 실행

const [openDate, setOpenDate] = useState(null); : 현재 열려있는 일기의 날짜, 수정 중일 경우 해당 날짜의 일기 내용을 폼에 렌더링

const [form, setForm] = useState({date: dayjs().format("YYYY-MM-DD"), content: "",}); : form은 일기 작성.수정 폼의 상태

- 일기 추가

setIsAdding(true): 새 일기 작성 모드 진입

setOpenDate(nulll): 기존 수정 모드 종료

-일기 삭제

delteteDiary: 삭제 여부를 confirm()으로 사용자에게 확인, deltet 연산자를 사용해 해당 날짜의 키 제거

-일기 책 클릭

handleBookClick: if: 이미 열려 있으면 닫기. setOpenDate(date) 새로 열기, setIsAdding(false) 추가 종료, setFrom 해낭 날짜의 내용 로딩

# 9) 구현 소감 (개발 과정에서 있었던 일, 느낌)

기능과 디자인까지 같이 구현하는 것이 너무 어렵기도 했고, 너무 많은 기능들을 넣으니 어떤 기능을 어떻게 쓰고 있는 지 헷갈려서 수정하고 업데이트 하는 것에 매우 어려움을 겼었습니다. 중간 중간 문법을 툴리거나 프롭스를 잘못하면 렌더링 자체가 되지 않아 안 되는 원인을 찾는 것이 매우 어려웠습니다. 조금 더 완벽한 페이지를 만들고 싶어서 아래 gpt참고 부분을 gpt로 해결한 점이 아쉬웠습니다. 시간이 조금 더 많았으면 직접 세세한 기능을 여러 사이트에서 공부해보고 구현을 해보면서 과정을 느껴보았을텐데, 시간이 촉박하여 수업시간에 배운 내용 외에 추가 기능은 gpt를 사용한 것이 너무 아쉬웠습니다. 이외의 기능과 페이지들은 직접 만들려고 노력하였습니다. Gpt를 사용한 부분은 소스를 더 분석해보고 공부해보고 싶다고 느끼게 되었습니다.

# 10) 참고 사이트, 출처 등 명기

ChatGPT를 사용한 부분(페이지의 완성도를 위하여 사용)

-dayjs라이브러리 사용

-일기 페이지 로컬스티리지에 json문자열로 저장방식

-캘린더 페이지

-tailWindcss를 이용한 디자인

-리뷰페이지의 별점 순으로 정렬기능