

Cal.log

Cal.log

이현호 | 유부미 | 이치우 | 정지현

Callog.COM

목차

01. 프로젝트 개요

주제 및 개발 목적, 타겟 사용자 분석, 비즈니스 및 마케팅 전략

02. 프로젝트 팀 구성 및 역할

팀원 별 프로젝트 담당업무

03. 프로젝트 수행 절차

플로우차트, 기술스택

04. 프로젝트 시연 및 기능소개

기능 소개

05. 자체 평가 의견

느낀점, Q&A

01

주제 및 개발 목적

프로젝트 주제

일정 기반 콘텐츠 작성 플랫폼

캘린더와 블로그 작성 기능을 통합한 웹 사이트

개발목적

콘텐츠 관리와 일정 계획의 통합

프로젝트 일정 및 데일리 일정의 구조적 관리

사용자 중심의 직관적 UI/UX 구현

타겟 사용자 분석

Why?

사용자가 꾸밀 수 있는 캘린더

상황에 맞게 변화하는 상태창

깔끔하고 직관적인 UI

딱딱하고 사무적인 디자인에서 벗어난
다이어리 스타일의 캘린더

사용자의 이익성

작업 흐름 통합: 일정과 콘텐츠를 하나의
플랫폼에서 관리 가능

집중력 향상: 일정과 글이 유기적으로 연결

사용 편의성: 블로그처럼 글을 쓰며,
캘린더로 관리 가능

맞춤 콘텐츠 관리: 데일리/ 프로젝트 일정에
맞춘 글 작성

페르소나

블로그 작성과 일정 관리 앱을 병행하는
사용자

20~30대 대학생 및 직장인 사용자

일정을 시각화하고 싶은 사용자

직관적인 글쓰기를 원하는 사용자

비즈니스 및 마케팅 전략

마케팅 전략

커뮤니티형 서비스로 확장 가능

일정 꾸준히 작성시 포인트 제공으로 유지율 증가 유도

콘텐츠 작성을 독려하는 주간 챌린지 -> SNS공유 유도

브랜드 네이밍으로 기억하기 쉬운 이름

수익화 방안

프리미엄 구독제(일정 반복 알림, 고급 템플릿, 다이어리 테마)

홈 화면 하단 배너 광고 노출로 수익 유입

유저가 직접 템플릿 제작 및 판매(수수료 수익)

브랜드 제휴(자기계발 도서, 생산성 앱 등과 제휴)

01

팀원 소개 및 역할 분담



팀장 이현호

팀장으로서 프로젝트 주도
일정 등록 및 관리 기능 구현
캘린더와 연동되는 모달창 구현



팀원 유부미

캘린더 라이브러리 및 공휴일 API를
활용한 캘린더 화면 구현



팀원 이치우

tanstack query를 활용한 데이터 캐싱,
패칭, 무한스크롤 구현 및 검색, 정렬,
태그별 조회 등 기타 편의 기능 구현



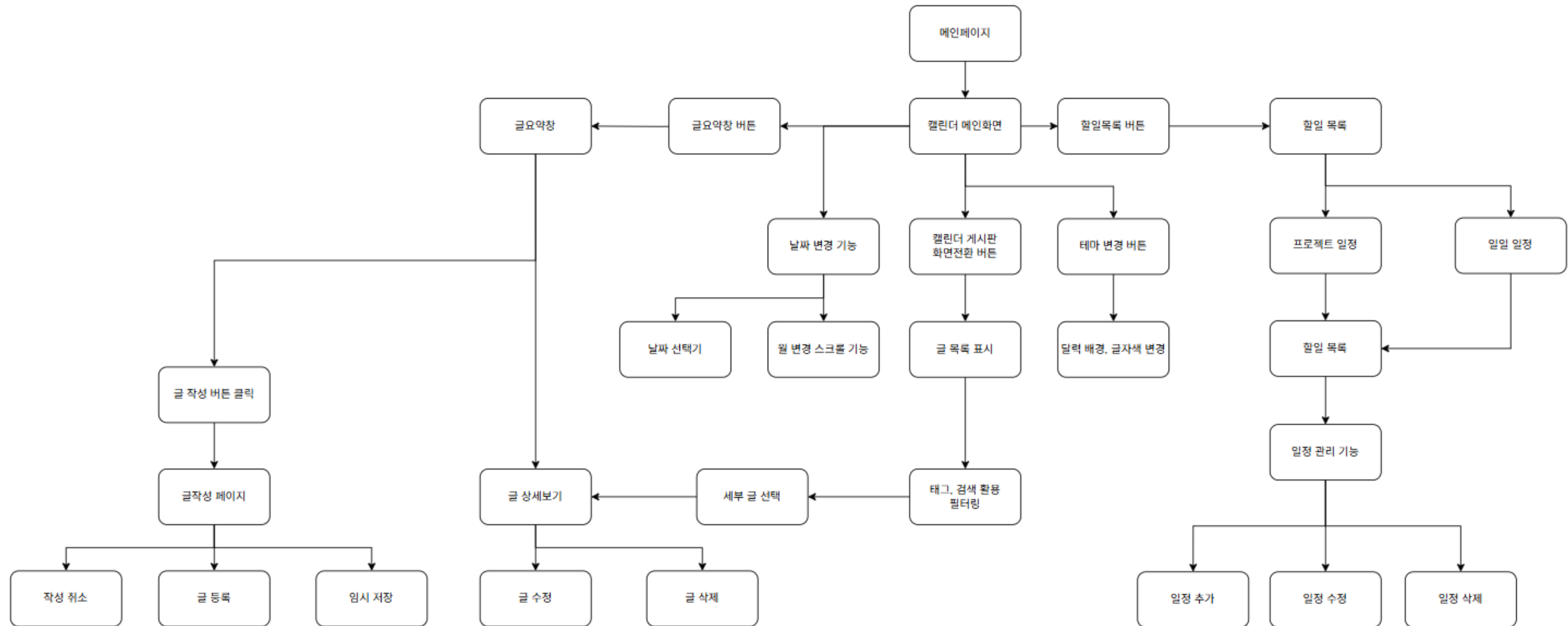
팀원 정지현

MarkDown Editor와 로컬 스토리지를
활용한 글 작성, 수정, 삭제 구현 및
UI/UX 설계

03. 프로젝트 수행절차

03

플로우 차트



01

기술 스택

개발 언어

HTML



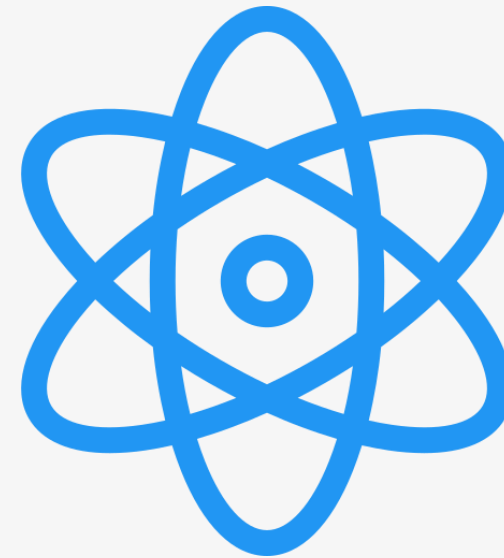
JS



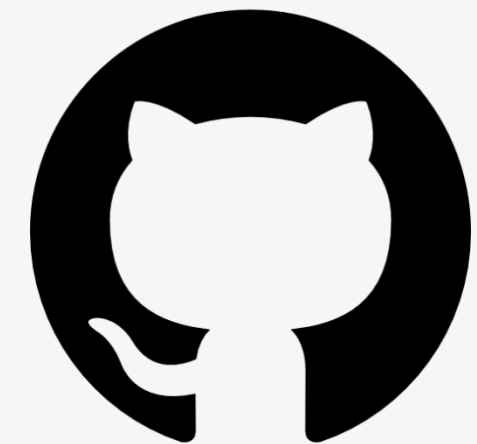
CSS



프레임워크



플랫폼 및 배포



React라이브러리

사용 도구

플랫폼

배포

Modal, Markdown, Calendar, TanStackQuery

공공데이터포털 API, Bootstrap

GitHub(<https://github.com/HyunHo-99/CalogProject.git>)

Vercel(<https://calog-project-hyunhos-projects-d89653df.vercel.app/>)

04. 프로젝트 시연 및 기능소개

01

기능소개(영상)



2025년 6월



Sun	Mon	Tue	Wed	Thu	Fri	Sat
01	02	03 임시공휴일(제21대 대...)	04	05	06 현충일	07
08	09	10 프로젝트 일정1	11	12	13	14
					프로젝트일정추가	
15	16	17	18	19	20	21
프로젝트일정추가						
22	23	24	25	26	27	28
프로젝트일정추가			+2	+1	+1	프로젝트 일정2
					+2	
29	30	01	02	03	04	05
프로젝트 일정2						
+1	+2	+2	+1			

01

기능 소개

캘린더 메인화면

Calendar 라이브러리 활용 캘린더 구현
공공데이터활용 공휴일API 활용 공휴일 구현
CSS 변수 활용 테마 구현
월 이동시 월 변동 구현

할일목록, 글요약 컴포넌트

Modal 라이브러리 활용 모달 **CRUD** 구현
캘린더 월 이동시 관련된 내용 나오도록 연동

게시판화면

TanStackQuery 라이브러리 활용 스크롤 이동시 최적화

글 상세 CRUD 구현

Markdown 활용 글 작성 페이지 구현
상세페이지에서 수정, 삭제 기능 구현

01

기능 소개

<Modal>

```
<ModalCreate
  isOpen={isOpenModal}
  onModal={closeModal}
  modalType={modalType}
/>
<ModalEdit
  isOpen={isEditorOpen}
  onModal={editModalClose}
  modalType={modalType}
  data={findData}
/>
<Modal
  isOpen={isOpen}
  onRequestClose={() => onModal(false)}
  contentLabel={getModalTitle()}
  ariaHideApp={false}
  className="modal_content"
  overlayClassName="modal_overlay"
>
```

<Markdown>

```
const useMarkdown = () => {
  return {
    code({ inline, className, children, ...props }) {
      const match = /language-(\w+)/.exec(className || "");
      return !inline && match ? (
        <SyntaxHighlighter
          style={oneDark}
          language={match[1]}
          PreTag="div"
          {...props}
        >
          {String(children).replace(/\n$/, "")}
        </SyntaxHighlighter>
      ) : (
        <code className={className} {...props}>
          {children}
        </code>
      );
    },
  };
};

<div className="markdown_preview">
  <h3>미리보기</h3>
  <ReactMarkdown components={markdown}>
    {content}
  </ReactMarkdown>
</div> /.markdown_preview
```

<Modal>

모달 라이브러리를 사용하여 생성창과, 수정창을
구분해 컴포넌트로 구현하고 각각의 필요한
구역에서 모달을 호출해서 사용

<Markdown>

마크다운 라이브러리를 사용하여 코드 블록이나
헤더, 굵은 글씨, 이탤릭 체와 같은
마크다운 문법을 지원합니다.
(작성, 상세, 수정 페이지에서 제공)

01

기능 소개

<공휴일 API>

```
allRequests.push(
  axios
    .get(url, { params })
    .then((response) => {
      const items = response.data.response.body.items?.item;
      if (!items) return [];

      const monthlyHolidays = Array.isArray(items) ? items : [items];

      return monthlyHolidays.map((h) => ({
        title: h.dateName,
        start: new Date(
          h.locdate
            .toString()
            .replace(/(\d{4})(\d{2})(\d{2})/, "$1-$2-$3")
        ),
        end: new Date(
          h.locdate
            .toString()
            .replace(/(\d{4})(\d{2})(\d{2})/, "$1-$2-$3")
        ),
        allDay: true,
        isHoliday: true,
      }));
    })
    .catch((err) => {
      console.error(`❌ ${year}년 ${month}월 공휴일 실패:`, err);
      return [];
    })
);
```

```
useEffect(() => {
  const fetchMergedEvents = async () => {
    const holidays = await getHolidayEventsByYears();
    setMergedEvents([...events, ...holidays]);
  };
  fetchMergedEvents();
}, [events]);
```

<공휴일 API>

컴포넌트에서 모달창에서 입력받은 **themeColor**을 통해 받은 값을 변수로 적용,
CSS파일에서 적용 시 **var**(전역변수) 사용을 통해 **Switch**문으로 컬러 변경시 원하는 구역에서 색상 변경 기능 구현

01

느낀점

이현호

프로젝트를 구상하면서 단순하고 배운기능을 통해 할 수 있는 것만 넣었다고 생각을 해서 쉽게 생각을 했지만, 팀이 구성되고 아이디어를 구상하는 과정에서 생각지 못한 기능들과 모두의 아이디어가 합쳐지면서 완성되는 과정이 즐거웠습니다.

하지만 역할분담 과정에서 밸런스를 잘 조절하지 못한 것 같기도 하였고, 페이지를 기준으로 전체 업무를 절반으로 나누다보니 반대편의 코드가 궁금해지는 경우가 생기기도 했습니다. 테마를 마지막에 구현해서 할일이 많이 생겼었는데 초반에 기능을 구현할 순서를 정하는 것이 중요하다고 생각했습니다.

정지현

처음 접해본 리액트로 기능 구현과 **CSS**까지 하면서 생각처럼 결과가 나오지 않는 어려움들을 겪었습니다. 서로 연결되는 페이지를 한 명이 아닌 여러명이 코드를 작성하다보니 처음 짜둔 순서대로 작성하지 못하고 다른 사람이 먼저 사용해야되는 코드들을 먼저 작성하게 되었습니다. 이로인해 처음 **WBS**를 작성할 때 어떤 부분이 겹치는지, 우선순위를 어떻게 주어야하는지를 알게되었습니다. 팀워크 면에서 서로 각자의 역할들을 빠르게 끝내서 병합하는 과정까지 큰 문제없이 진행되었고 각자의 의견들을 활발하게 제시하는 점에서는 좋았지만 각자 원하는 점들이 있어 의도치않게 의견이 부딪치게 되었던 점이 아쉬웠던 것 같습니다

유부미

리액트 수업을 들을 때만 해도 “과연 실제 프로젝트를 제대로 해낼 수 있을까?” 하는 걱정이 있었지만, 막상 개발에 들어가 보니 어렵게 느껴졌던 개념들이 직접 구현을 통해 하나씩 체득되기 시작했습니다. 특히 라이브러리 기능과 컴포넌트 단위의 구조 분리를 직접 해보면서 실전에서만 익힐 수 있는 개발 감각을 얻을 수 있었습니다. 다만, 초반 진행이 순조로웠던 탓에 일정 계획을 다소 여유 있게 잡았고, 마감 직전에 예상치 못한 문제를 마주했을 때는 대응 시간이 부족했던 점이 아쉬웠습니다. 다음 프로젝트에서는 일정 계획을 조금 더 타이트하게 구성하여 더 높은 완성도로 마무리할 수 있도록 개선하고 싶습니다.

이치우

저번 자바 프로젝트를 하면서 가장 큰 문제가 됐던 부분이 병합 과정이었는데 이번 프로젝트에서는 다행히 팀장님께서 머지를 잘 해주셔서 큰 문제없이 진행되었습니다. 프로젝트를 진행하며 다양한 리액트 훅들에 익숙해지고 프론트엔드의 기초적인 기술들을 익히게 된 점도 좋았습니다. 다만 테마 변경 기능을 분명히 초반에 구상해두었음에도 신경을 쓰지 않고 **html/css**를 대충 짜다 보니 마지막에 테마 변경을 구현할 때 구현하기 어렵지 않은 난이도임에도 불구하고 많은 시간이 소모되었던 점은 아쉬웠습니다.

02

업그레이드 방안

일정공유

캘린더를 같이 볼 수 있도록 초대 및 링크 기능을 통해
캘린더 내용을 공유하는 기능 추가

템플릿 공유 기능

사용자 별 캘린더 꾸미기 기능으로 테마 공유 기능 추가

튜토리얼 기능

최초 웹사이트 방문시 튜토리얼을 통해 세부 기능 설명,
마크다운 관련 간략한 문법 설명 및 도움말 기능 추가

알림 기능

일정별 기간 도래 하루 전, 일주일 전 등
사용자가 원하는 시기에 알림표시
프로젝트 일정의 경우 **D-day**에 따라 시각적 효과 구현

02

QnA

