

주간 보고서 시스템 만들기

파일 경로

C:\project

🏗️ 전체 시스템 구조

기능 블록	설명
공지 블로그	파일 업로드, 글 작성 기능 포함. 교수님과 본인만 접근 가능 (관리자 권한).
전체 공지사항	모든 사용자(팀원 등)가 읽을 수 있도록 교수님과 본인이 작성하는 공지글 게시 판.
최신 IT 뉴스 블로그	최신 IT 뉴스 모음 블로그, 누구나 접근 가능. 뉴스 수집은 수동 작성 또는 크롤링 자동화 가능.

주간 보고서 시스템은 대략적으로 이렇게 만드려고 한다.

✨ 주요 기능별 세부 설계

1 공지 블로그 (관리자 전용)

- **파일 업로드:** multer 같은 Node.js용 미들웨어 사용 → 서버 저장 or 클라우드 스토리지 연동.
- **글 작성/편집:** Markdown 또는 WYSIWYG 에디터 (예: Toast UI Editor) 제공.

- **접근 제한:** 관리자만 접근 가능, 라우팅 및 API 요청 시 토큰 인증 필수.

2 전체 공지사항

- **글 작성/편집:** 공지 등록 시 자동으로 전체 사용자에게 알림 가능 (예: 이메일, 앱 내 알림).
- **공개 범위:** 모든 로그인 사용자가 볼 수 있도록 공개.
- **작성자 표시:** 글마다 작성자(교수님/본인) 정보 노출.

3 최신 IT 뉴스 블로그

- **뉴스 작성:** 직접 수동 작성 또는 뉴스 사이트 크롤링 후 자동 등록.
- **크롤링 자동화 (선택):** Node.js에서 puppeteer, cheerio 등으로 뉴스 사이트 스크랩.
- **SEO 대응:** Next.js 사용 시 정적 페이지 생성(SSG)로 검색엔진 최적화.

권한 및 접근 관리 설계

사용자 유형	권한
관리자 (교수님/본인)	공지 블로그 작성/수정/삭제, 전체 공지 작성/수정/삭제, IT 뉴스 관리
일반 사용자	전체 공지 읽기, IT 뉴스 읽기
비로그인 사용자	IT 뉴스 읽기 (전체 공지과 공지 블로그는 접근 차단)

✅ **중요:** 관리자 전용 영역은 반드시 인증 체크 (JWT 토큰 or 세션 기반)로 보호해야 하고, 관리자 UI는 관리자 계정에서만 접근 가능하도록 분리해야 합니다.

최종 기술 스택 (PostgreSQL 기반)


영역	선택 기술	설명/비고
프론트엔드	Next.js (React 기반)	SEO + SSG/SSR 지원 → IT 뉴스 블로그에 유리. Vercel 무료 배포 가능.
UI 라이브러리	Tailwind CSS(필요시 선택)	빠르고 유연한 스타일링. shadcn/ui로 React 컴포넌트 라이브러리 활용.
상태 관리	React Context, React Query (또는 SWR)	관리자용 데이터, 로그인 상태, API 연동 관리.
백엔드	Node.js + Express	빠르고 간단한 REST API 서버. 초기에 NestJS까지는 필요 없음.

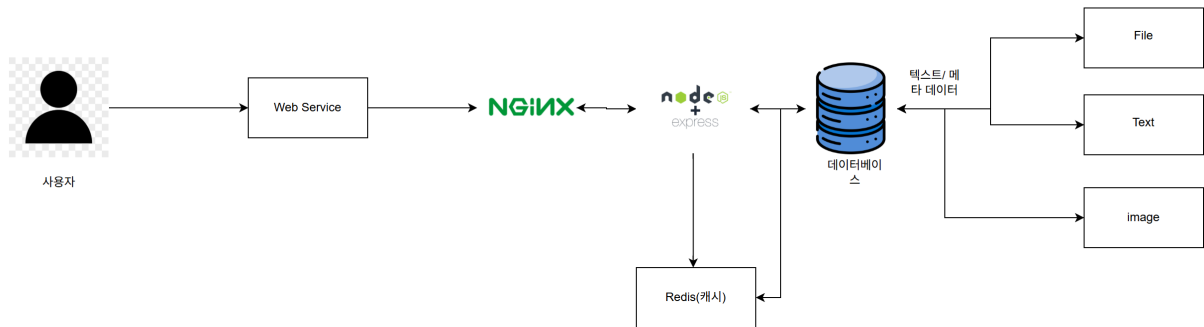
파일 업로드	multer (Express 미들웨어) + 서버 로컬 저장	작은 규모에서 서버의 <code>uploads/</code> 폴더로 파일 저장. 나중에 AWS S3로 교체 준비할 수 있도록 코드 분리.
인증/보안	JWT (jsonwebtoken) + bcrypt	관리자/사용자 권한 분리. 비밀번호 해시화. 토큰 기반 인증.
데이터베이스	PostgreSQL + Prisma (ORM)	Supabase 무료 PostgreSQL 사용 가능. Prisma로 타입 안전 쿼리 작성, DB 마이그레이션 관리.
배포 (프론트)	Vercel	Next.js 앱 무료 배포. HTTPS 기본 적용.
배포 (백엔드)	Render, Railway (혹은 Fly.io)	Node.js 서버 무료 배포. PostgreSQL 연결도 함께 제공됨 (Supabase와 연동하거나 Render 자체 DB 사용 가능).
DB 호스팅	Supabase (PostgreSQL)	무료 플랜에서 일정 용량 제공. 관리 UI, 인증, 스토리지 등 부가 기능도 있음.
도구	GitHub (코드 관리), Vercel CLI, Prisma Studio, Postman	GitHub로 버전 관리. Vercel CLI로 배포. Prisma Studio로 DB 시각화 관리. Postman으로 API 테스트. —

다이어그램 전체 개요

전체 아키텍처 설계.drawio

draw.io is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams

 https://viewer.diagrams.net/?border=0&tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=%EC%A0%84%EC%B2%B4%20%EC%95%84%ED%82%A4%ED%85%8D%EC%B2%98%20%EC%84%A4%EA%B3%84.drawio&dark=auto#R%3Cmxfile%3E%3Cdiagram%20name%3D%22ED%8E%98%EC%9D%B4%EC%A7%80-1%22%20id%3D%22QCIG3aPFW



1 사용자 > Web Service

- 사용자 (브라우저) 클라이언트 (사용자)가 웹 브라우저나 모바일을 통해 시스템에 접근합니다.
- Web Service
Next.js 같은 프론트엔드 앱 (React 기반)으로 작성됩니다.
→ 사용자에게 UI를 제공하고, 필요할 때
NGINX나 백엔드로 API 요청을 보냅니다.

2 NGINX (리버스 프록시)

- 요청은 NGINX로 먼저 들어옵니다.
 - SSL/TLS (HTTPS) 처리.
 - 정적 파일 캐싱 (필요할 경우).
 - /api 요청은 Express로, 그 외는 프론트엔드로 라우팅.

3 Express (Node.js API 서버)

- 백엔드 API 서버.
 - REST API 제공.
 - 파일 업로드, 공지글 작성, IT 뉴스 관리 같은 주요 비즈니스 로직 담당.

→ DB 접근 및 Redis 캐시 접근 포함.

4 데이터베이스 (PostgreSQL 등)

- 서비스의 중심 데이터 저장소.
 - 사용자 정보, 공지글, 뉴스 데이터, 파일 메타데이터 저장.
 - **주의:** 실제 파일 본체(File/Image)는 보통 DB가 아닌 로컬 디스크나 외부 스토리지에 저장.
-

5 Redis (캐시)

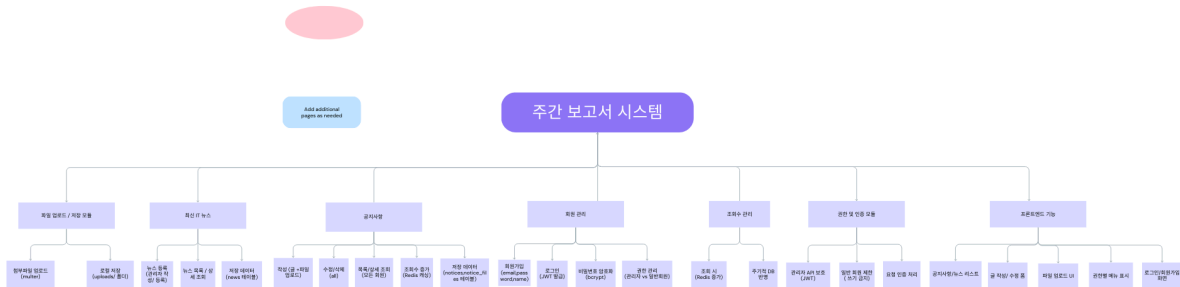
- Express 서버의 요청 처리 속도를 높이기 위한 인메모리 캐시.
 - 예: 자주 요청되는 공지글 리스트, 인기 뉴스, 사용자 세션 정보.
 - Express는 DB 조회 전에 Redis에서 먼저 데이터를 조회하여 속도를 최적화.
-

6 File/Text/Image

- DB에서 관리되는 데이터 타입.
 - **File:** 업로드된 파일(실제 데이터는 로컬 uploads/ 또는 S3, 메타데이터는 DB에).
 - **Text:** 공지사항, 뉴스 본문 등 텍스트 데이터.
 - **Image:** 이미지 파일 (실제 데이터는 로컬/S3, 메타데이터는 DB에).

현재 다이어그램에서는 DB에서 세 가지로 분기되어 있지만,

실제 아키텍처에서는 Express가 요청을 처리하고, 그 요청에 따라 DB 또는 파일 시스템에서 데이터를 불러옴.



→ 모든 회원이 최신 IT 뉴스 목록 및 상세 내용 열람.

- **저장 데이터 (news 테이블)**

→ 뉴스 제목, 요약, 링크, 작성자 정보 등을 DB에 저장.

◆ 3 공지사항

✓ 하위 기능:

- **작성 (글 + 파일 업로드)**

→ 관리자만 공지사항 작성 가능, 글과 첨부파일 함께 등록.

- **수정/삭제 (관리자만)**

→ 작성된 공지사항을 관리자만 편집·삭제 가능.

- **조회수 증가 (Redis 캐시)**

→ 공지사항 조회 시 Redis로 캐싱하여 빠른 조회수 처리.

- **저장 데이터 (notices, notice_files)**

→ 공지사항 본문과 파일 메타데이터 저장.

◆ 4 회원 관리

✓ 하위 기능:

- **회원가입 (email, password, name)**

→ 신규 회원 등록 기능.

- **비밀번호 암호화 (bcrypt)**

→ 사용자 비밀번호를 안전하게 해시 처리.

- **로그인 (JWT 발급)**

→ 로그인 시 JWT 토큰 발급, 이후 요청에 인증 사용.

- **비밀번호 인증/복호화 (bcrypt)**

→ 로그인 시 입력값을 해시값과 비교하여 인증.

◆ 5 조회수 관리

✓ 하위 기능:

- **조회 시 (Redis 증가)**

→ 사용자가 글을 읽을 때 Redis에서 실시간 조회수 관리.

- **주기적 DB 반영**

→ 일정 주기마다 Redis 데이터를 DB로 업데이트.

◆ 6 권한 및 인증 모듈

✓ 하위 기능:

- **관리자 API 보호 (JWT)**

→ 관리자 전용 API는 JWT 권한 검사 후 접근 허용.

- **일반 회원 제한 (쓰기 금지)**

→ 일반 회원은 열람만 가능, 쓰기/삭제는 차단.

- **요청 인증 처리**

→ 클라이언트 요청 시 Authorization 헤더로 토큰 검증.

◆ 7 프론트엔드 기능

✓ 하위 기능:

- **공지사항/뉴스 리스트**

→ 사용자에게 최신 글 목록을 보여줌.

- **글 작성/수정 폼**

→ 관리자용 글 작성·편집 화면.

- **파일 업로드 UI**

→ 프론트에서 첨부파일 올리는 인터페이스.

- **권한별 메뉴 표시**

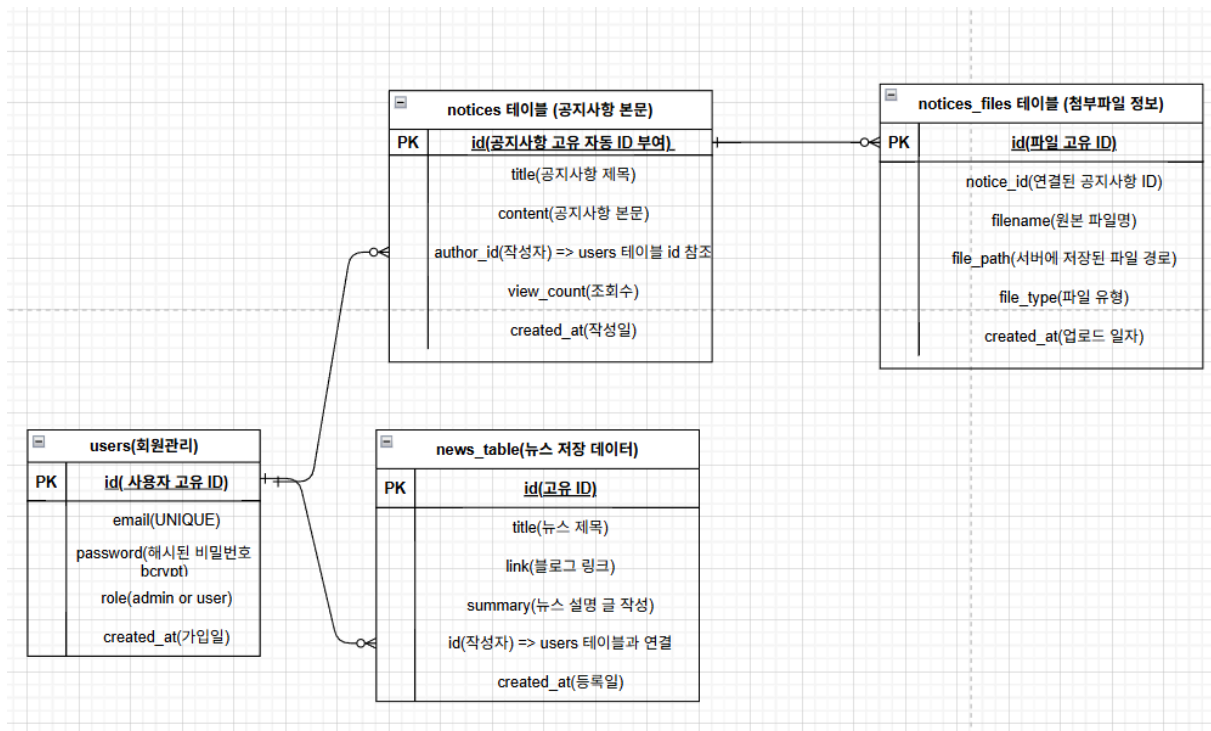
→ 로그인/권한에 따라 메뉴 다르게 표시.

- **로그인/회원가입 화면**

→ 사용자 인증용 화면.

ERD 설계

<https://drive.google.com/file/d/1XzEE4mk3pOi9VUh024w659epV2r-IGLk/view?usp=sharing>



1 users (회원 관리)

컬럼명	키 설계
id	PK (Primary Key, Auto Increment)
email	UNIQUE (이메일 중복 불가)
password	일반 컬럼
role	일반 컬럼 (ENUM 또는 VARCHAR 추천)
created_at	일반 컬럼
gender(성별)	일반 컬럼

✓ 포인트:

- **email** 에 UNIQUE 추가 → 동일 이메일 중복 가입 방지.
- **role** ENUM(admin, user) 설정하면 데이터 무결성 강화.

2 news_table (뉴스 저장 데이터)

컬럼명	키 설계
id	PK (Primary Key, Auto Increment)
title	일반 컬럼 (중복 허용 가능)
link	일반 컬럼

summary	일반 컬럼
author_id	FK → users(id) (작성자 연결)
created_at	일반 컬럼

✓ **포인트:**

- `author_id` 외래키로 users 테이블 연결.
- `link` 는 중복 허용할지 고민 → 필요하면 UNIQUE 추가 가능.

3 notices (공지사항 본문)

컬럼명	키 설계
id	PK (Primary Key, Auto Increment)
title	일반 컬럼 (중복 허용)
content	일반 컬럼
author_id	FK → users(id) (작성자 연결)
view_count	일반 컬럼 (조회수, 기본값 0)
created_at	일반 컬럼

✓ **포인트:**

- `author_id` 외래키로 users 테이블 연결.
- 제목에 UNIQUE 필요 없다면 중복 허용.

4 notice_files (첨부파일 정보)

컬럼명	키 설계
id	PK (Primary Key, Auto Increment)
notice_id	FK → notices(id) (공지사항 연결)
filename	일반 컬럼 (업로드된 원본 파일명)
file_path	일반 컬럼 (저장 경로)
file_type	일반 컬럼
created_at	일반 컬럼

✓ **포인트:**

- `notice_id` 외래키로 notices 테이블 연결.
- `file_path` 는 서버 경로 기준으로 UNIQUE 지정하면 덮어쓰기 방지 가능.

주간 보고서 시스템 개발

—