성균관대학교 LCDA 합주 캘린더 구현 계획

노션 감성의 월간 보드와 구글 캘린더 스타일의 주간 타임라인을 결합해 성균관대학교 LCDA 합주 일정을 체계적으로 관리하는 Next.js 애플리케이션을 구축하기 위한 실행 계획이다. 사용자 경험, 데이터 모델링, 인프라, 배포까지 전 과정을 단계적으로 설명한다.

# 프로젝트 목표

LCDA 합주 스케줄링을 위한 단일 페이지 애플리케이션을 구축해 팀원 모두가 쉽게 일정 생성·공유·수정할 수 있는 환경을 마련한다.

* Notion 캘린더의 미니멀 UI 흐름과 Google 캘린더의 시간 블록 인터랙션을 조합
* 30분 단위 슬롯 기반의 반복 일정/개별 일정 관리
* 데스크톱 퍼스트 경험을 제공하고, 추후 모바일 대응성을 확장할 수 있도록 설계

# 주요 요구사항 정리

* 월간 뷰: `Calender/notion\_cal.png` 레이아웃을 참고해 블록형 날짜 카드, 상단 주간 헤더, 노션 스타일 서체·컬러 적용
* 주간 타임라인 뷰: `Calender/google\_cal.png`를 참고해 30분 간격 세로 그리드와 좌측 시간 스케일을 구성
* 오늘 날짜 표기: 월간 뷰에서는 붉은 원형 하이라이트, 타임라인 뷰에서는 붉은 수평 라인과 타임 칩으로 표시
* 날짜 셀 Hover 시 + 버튼이 150ms 페이드 & 1.1배 스케일로 등장, 클릭 시 일정 생성 모달 호출
* + 버튼 모달: 제목, 설명, 장소, 색상, 시작/종료 시각 (30분 단위), 반복 요일 선택, 반복 종료 조건(횟수/날짜)을 입력
* 반복 일정은 선택한 요일 기준으로 생성되며 지정 횟수만큼 월간/주간 뷰에 표시, 단일 일정은 해당 일자에만 노출
* 이벤트 클릭 시 상세 패널이 슬라이드 인되어 세부 정보, 수정, 삭제를 처리
* 캘린더 빈 공간 더블클릭: 뷰 전환 토글 (월간 ↔ 주간). 전환 시 프레이머 모션을 활용한 페이드 + 스케일 전환 애니메이션 적용
* 주간 뷰 시간 슬롯 클릭: 해당 날짜/시간이 사전선택된 상태로 일정 생성 모달 오픈
* 데이터는 서버에 저장되어 새로고침/다중 디바이스 간 동기화가 가능해야 함

# 기술 스택 및 인프라

* Next.js 14 (App Router) + TypeScript 기반 SPA/SSR 하이브리드 구성
* 스타일: Tailwind CSS + custom CSS 변수로 노션/구글 색상 팔레트 정합, 다크 모드는 후속 단계 고려
* UI 컴포넌트: shadcn/ui 프리셋을 커스터마이징해 입력·모달 일관성 확보
* 상태 관리: TanStack Query로 서버 상태를 캐싱, Zustand로 뷰 토글·모달 등 클라이언트 UI 상태 관리
* 폼 관리: React Hook Form + Zod로 유효성 검증 로직 통일
* 애니메이션: Framer Motion으로 뷰 전환, Hover + 버튼 등장, 모달 트랜지션 처리
* 데이터 계층: Prisma ORM + Supabase(PostgreSQL) 연동, 로컬 개발은 Docker 기반 PostgreSQL 또는 Prisma SQLite 프리셋 사용
* 실시간 동기화는 1차 범위에서 제외, 필요 시 Supabase Realtime 또는 Pusher를 추후 연동
* 배포: Vercel + Supabase (Managed DB), 환경 변수는 Vercel env manager로 관리

# 시스템 아키텍처

App Router를 활용해 UI 레이어와 서버 로직을 분리하고, Prisma를 통해 데이터베이스와 통신한다.

* 라우팅 레이어: `/` 월간 뷰, `/week` 주간 뷰, 서버 구성 요소를 활용해 초기 데이터를 스트리밍
* 서버 액션/API 라우트: `/app/api/events`에서 CRUD 및 반복 일정 계산을 처리
* 서비스 레이어: 일정 생성 시 비즈니스 규칙(30분 스냅, 반복 룰 적용)을 한 곳에서 검증
* 클라이언트 레이어: 뷰 전환, 모달, Drag & Drop(후속 단계) 등 인터랙션 관리

# 데이터 모델

반복 규칙을 별도 테이블로 분리해 단일 이벤트와 반복 이벤트를 동일한 구조에서 관리한다.

* Event: id (UUID), title, description, location, color, start\_datetime (tz aware), end\_datetime, timezone, recurrence\_rule\_id, created\_at, updated\_at
* RecurrenceRule: id, frequency (weekly), interval, weekdays (int array), count (nullable), until (nullable date), timezone
* EventException (Optional 2차): id, event\_id, date, is\_cancelled, override\_start, override\_end
* ViewPreference (Optional): 사용자별 기본 뷰, 필터를 저장하는 객체
* Prisma schema에서 DateTime(UTC) + timezone 문자열을 병행해 오프셋 손실을 방지

# 화면 및 라우팅 구조

* 레이아웃 컴포넌트: 상단 네비게이션(로고, 뷰 토글, 날짜 이동, 반복 보기 필터), 우측 유틸(설정, 도움말)
* 월간 뷰 컴포넌트: MonthGrid, DayCell, EventBadge, HoverAddButton
* 주간 뷰 컴포넌트: TimelineGrid, TimeColumn, TimelineEventCard
* 모달/패널: EventFormModal, EventDetailSheet, RepeatRuleConfigurator
* 설정 페이지(후속): 반복 규칙 프리셋, 테마 설정

# UI/UX 세부 설계

Notion과 Google 캘린더의 시각적 요소를 혼합하되 전체 톤앤매너를 통일한다.

* 타이포그래피: 기본 `Inter` (Notion), 숫자/시간 영역에 `Roboto Mono`를 혼합해 가독성 확보
* 색상 팔레트: 노션의 Warm Gray 계열 배경 + 구글 캘린더의 파스텔 강조 컬러를 Tailwind theme에 매핑
* 카드 섀도우: 노션식 낮은 대비 그림자, Hover 시 음영을 살짝 강화
* 이벤트 칩: 카테고리 별 색상 태그, 텍스트 대비도 4.5:1 이상 유지
* 반응형: 1280px 기준 레이아웃, 1024px 이하에서는 Week 뷰를 가로 스크롤 허용, 모바일 최적화는 단계2 계획

# 상호작용 흐름

* DayCell Hover → + 버튼 페이드인 → 클릭 시 EventFormModal 오픈
* EventFormModal 저장 → 서버 액션으로 유효성 검증 → Prisma create/update → Query invalidate → UI 갱신
* Event 클릭 → EventDetailSheet가 우측에서 slide-in, 수정 버튼으로 FormModal 재사용
* Calendar double-click → view state 토글 → Framer Motion layout animation으로 전환
* 주간 뷰 시간 블록 드래그(훅업션) → 시작/종료 자동 채움 (1차에서는 클릭 & 조정으로 제한)

# 일정 생성 모달 설계

* 기본 정보: 제목(필수), 설명, 장소, 색상
* 시간 선택: 시작/종료 TimePicker는 30분 단위, 종료 시간은 시작 이후만 선택 가능
* 반복 설정: 요일 멀티셀렉트(Toggle Button Group), 반복 간격(주 단위), 종료 옵션(미지정, N회, 특정 날짜)
* 추가 옵션: 알림, 메모, 담당자 태그는 후속 릴리즈로 예약
* 검증 실패 시 인라인 에러와 상단 요약 표시

# 반복 일정 처리 전략

RRULE 로직을 직접 구현하되, `rrule` 패키지를 활용해 서버에서 반복 인스턴스를 계산한다.

* 기본 이벤트는 start\_datetime 기준, 선택한 요일에 맞춰 가장 가까운 날짜로 생성
* API 응답 시 요청 구간(start, end)에 해당하는 반복 인스턴스를 계산해 반환
* 반복 종료 조건(count 또는 until)을 순회하면서 캐싱 없이 즉시 연산 (월간/주간 범위는 제한적이므로 성능 양호)
* EventException을 활용해 단일 회차 취소/시간 변경을 추후 지원

# 상태 관리와 동기화

* TanStack Query Key: `["events", view, startISO, endISO]` 구조로 범위별 데이터를 캐시
* 모달 폼은 Zustand store로 open/close state, 선택된 날짜·시간을 공유
* 뷰 전환 시 공유 컨텍스트를 통해 선택된 날짜를 유지 → 월간/주간 뷰 모두 동일 anchor date 사용
* Optimistic update: 생성 시 임시 이벤트를 렌더링하고 실패 시 롤백

# 접근성 및 국제화

* Keyboard Navigation: 화살표 키로 날짜 이동, Enter로 모달 오픈, ESC로 닫기
* ARIA 속성: 모달 접근성, 이벤트 리스트에 role="listitem" 적용
* 언어: 한국어 UI를 기본으로 하되, i18n 준비를 위해 `next-intl` 또는 `@lingui` 도입 고려
* 색 대비 및 폰트 크기를 WCAG AA 수준으로 맞추기

# 배포 및 무료 도메인 전략

Vercel 배포를 기본으로 하고, 무료 도메인을 연결해 LCDA 구성원이 손쉽게 접근하도록 구성한다.

1. GitHub 저장소와 Vercel 연동 → main 브랜치 push 시 자동 배포 파이프라인 설정
2. 환경 변수(SUPABASE\_URL, SUPABASE\_KEY 등)는 Vercel Project Settings에서 관리
3. 무료 도메인: Freenom(.tk/.ml 등) 또는 eu.org에서 `lcda-ensemble.tk`와 같은 도메인 등록
4. Freenom DNS 설정에서 Vercel 네임서버(ns1.vercel-dns.com, ns2.vercel-dns.com)로 변경
5. Vercel Domain 탭에 도메인 추가 → Automatic SSL (Let’s Encrypt) 활성화
6. 도메인 등록이 지연될 경우를 대비해 기본 `lcda-ensemble.vercel.app` 서브도메인을 안내

# 개발 로드맵 (3 스프린트 제안)

각 스프린트는 1~2주를 가정하며 우선순위 기능부터 단계적으로 완성한다.

* 스프린트 1: 프로젝트 세팅, 디자인 토큰, 월간 뷰 기본 렌더링, 이벤트 생성/저장 (단일 일정)
* 스프린트 2: 반복 규칙 구현, 주간 타임라인 뷰, 뷰 전환 애니메이션, 상세 패널/수정/삭제
* 스프린트 3: 스타일 디테일 다듬기, 접근성 개선, 배포 자동화, 무료 도메인 연결, QA 및 문서화

# 테스트 및 품질 보증

* 단위 테스트: 날짜 계산, 반복 규칙, 시간 스냅 함수는 Vitest로 커버
* 통합 테스트: Next.js Route Handler를 Supertest로 호출해 이벤트 생성/수정 플로우 검증
* E2E 테스트: Playwright로 주요 사용자 시나리오 (이벤트 생성, 반복 설정, 뷰 전환) 자동화
* 시각적 회귀: Chromatic 또는 Percy를 고려해 주요 화면 변화 추적 (후속)

# 향후 확장 아이디어

* 모바일 최적화 및 PWA 지원으로 오프라인 확인
* 알림 및 캘린더 구독(ICS export, Google Calendar Sync)
* 역할 기반 권한 (운영진/단원), 일정 승인 워크플로우
* 실시간 동기화 및 공동 편집 기능

# 아이콘 및 시각 자산 가이드

노션의 심플함과 구글 캘린더의 친숙한 톤을 유지하기 위해 라인 아이콘과 미세한 채도 컬러를 조합한다.

* 아이콘 세트: Phosphor Icons Thin/Light를 기본으로 사용, Stroke 1.5px, 모서리는 2px 라운딩
* 기본 색상: 비활성은 #8E8E8E, 활성/Primary 상태는 Tailwind `sky-500`, 경고/삭제는 `rose-500`
* 버튼 내 아이콘 크기: 16px (액션), 20px (네비게이션), 24px (Floating Add)
* 아이콘 주변 패딩: 최소 8px, Hover 시 배경에 6% 투명도의 primary tint 적용
* 애니메이션: Hover 150ms ease-out opacity + scale(1.05), 활성화 시 200ms 컬러 전환
* Accessibility: 아이콘만 존재하는 버튼에는 `aria-label` 지정, Tooltip (Radix) 300ms 지연 노출
* 네비게이션 & 뷰 전환
* 오늘로 이동: `calendar-check` 아이콘, 네비게이션 바 좌측. 클릭 시 anchor date를 현재 날짜로 이동.
* 이전/다음 주·월: `caret-left`, `caret-right`, 하이라이트 링 버튼. 연속 클릭 대비를 위해 Hover 시 배경 밝기 +8%.
* 뷰 토글: 월간/주간 버튼 옆 `arrows-left-right` 아이콘을 배치해 더블클릭 전환 힌트 제공.
* 이벤트 CRUD
* 추가: DayCell Hover 시 `plus` 아이콘이 원형 버튼으로 등장 (36px), 배경은 #1A73E8 16% 투명, 아이콘은 실선 흰색.
* 수정: EventDetailSheet 헤더 우측 `pencil-simple` (18px), Hover 시 #0B57D0.
* 삭제: 동일 위치 `trash-simple`, Hover 시 붉은 백그라운드(#FEE2E2) + 아이콘 #DC2626.
* 반복 설정: FormModal 내 요일 선택 헤더에 `arrows-clockwise` 아이콘을 두어 반복 옵션 시각화.
* 정보 & 상태
* 일정 상세: 상세 패널 하단 `info` 아이콘과 함께 메타데이터(작성자, 반복 규칙)를 표시.
* 반복 표식: MonthGrid에서 반복 일정은 이벤트 칩 좌측에 8px 루프 배지(배경 #BAE6FD)를 추가.
* 더블클릭 안내: 빈 캘린더 영역 오른쪽 하단에 `cursor-click` 유도 아이콘을 연한 회색(#D1D5DB)으로 배치.

# 핵심 컴포넌트 비주얼 사양

각 뷰와 패널이 어떤 레이아웃과 톤으로 구성되는지 세부적으로 정의한다.

* TopNavigation
* 높이 64px, 배경 #F8FAFC, 하단 1px 분리선(#E2E8F0). 좌측에 LCDA 로고(텍스트), 우측에 뷰 토글/검색/사용자 메뉴 정렬.
* 뷰 토글은 Pill 버튼(높이 36px, Border radius 9999px), 활성 탭은 흰색 배경 + 그림자, 비활성은 투명 테두리.
* 검색 입력은 아이콘 인풋(아이콘 #94A3B8) + placeholder "팀원 찾기/연습실 예약".
* MonthGrid
* 7열×5~6행 CSS Grid, 각 셀은 1:1.05 비율. 배경 #FFFFFF, 카드 섀도우 0 8px 24px -20% 투명도.
* 날짜 헤더는 상단 좌측 12px 여백, 숫자 폰트 14px SemiBold. 오늘 날짜는 20px 원형 배지(배경 #EF4444, 텍스트 흰색).
* 이벤트 칩은 최대 3줄 표시, 초과 시 `+N` 더보기 라벨. 칩 높이 24px, Border radius 6px, 좌측 4px 컬러 Strip.
* HoverAddButton은 셀 우측 하단 고정, opacity 0→1 전환. 클릭 영역을 40px로 확대해 접근성 확보.
* WeekTimeline
* 좌측 72px 시간 컬럼(배경 #F1F5F9), 나머지 영역은 7열 플렉스. 30분 간격을 48px 높이로 렌더링.
* 현재 시간 라인은 2px 솔리드 #F44336, 우측 끝에 시간 칩(12px Roboto Mono)을 붙여 가독성 강화.
* 타임블록은 드래그/리사이즈 핸들(상하단 4px 영역)과 이벤트 색상 배경, 70% 투명도 Hover 시 90%.
* EventDetailSheet
* 우측에서 320px 폭으로 슬라이드 인, 배경 흰색, 상단에 일정 색상 스트립(높이 4px).
* 헤더에는 제목(18px Bold) + 반복 배지 + Close 아이콘. 본문은 두 섹션(기본 정보, 반복 정보)으로 나눔.
* Action Row는 Ghost 버튼(수정, 삭제) + Primary 버튼(일정 복제) 배치.
* EventFormModal
* 크기 640px×자동, 라운딩 20px, 상단 그림자. 배경 Blur 12px로 모달 집중 효과.
* Form은 2컬럼 레이아웃: 좌측 기본 정보, 우측 반복/세부 옵션. 모바일 브레이크포인트에서 1컬럼으로 전환.
* 요일 선택은 Segmented Control 형태, 선택 시 배경 #0EA5E9, 텍스트 흰색.

# 상태 및 피드백 디자인

* 로딩: MonthGrid/WeekTimeline 상단에 3칸 Skeleton shimmer(1200ms).
* 빈 상태: "아직 일정이 없어요" 메시지와 `calendar-plus` 아이콘, Primary 버튼으로 첫 일정 추가 CTA 제공.
* 오류 상태: 상단 Toast(배경 #FEE2E2, 아이콘 `warning-circle`)와 재시도 버튼, 반복적으로 실패할 경우 고객지원 링크 노출.
* 성공 알림: 우측 상단 Toast(배경 #DCFCE7, 아이콘 `check-circle`) 2초 노출, 접근성을 위해 화면리더 friendly 메세지 포함.
* 모달 validation 실패: 필드 하단 붉은 보더 + 12px 안내문, 상단 summary에 error 아이콘과 메시지 노출.

# 1차 구현 진행 상태

Next.js 14 기반 초기 캘린더 애플리케이션 구조를 구현하고 개발 환경을 가동 중이다. 현재 단계의 UI/로직 구현 내용을 요약한다.

* Tailwind CSS 테마 토큰 구축: `globals.css`에 노션×구글 팔레트, radius, shadow 변수 정의
* AppShell 헤더: 범위 이동, 뷰 토글, 오늘로 이동, 일정 작성 CTA 및 안내 아이콘 구현
* 월간 뷰(노션 레퍼런스): 날짜 카드 Hover + 버튼, 오늘 배지, 반복 일정 배지, 더보기 인터랙션까지 구현
* 주간 뷰(구글 레퍼런스): 30분 슬롯 렌더링, 현재 시간 라인, 클릭 시 모달 오픈 훅 연결
* Radix Dialog 기반 Event Composer 모달: 반복 옵션/요일 토글/반복 종료 로직 UI 구성
* Framer Motion을 이용한 월간↔주간 전환 애니메이션 및 더블클릭 트리거 적용
* 샘플 일정 Seed 데이터 삽입, 보기 전환 시 범위별 필터링 처리
* Lint 통과 (`npm run lint`), 개발 서버 `npm run dev -- --hostname 0.0.0.0 --port 3000`로 실행 중 (PID 기록)

**서버 실행 정보:** http://127.0.0.1:3000 (PID 1690771, 2025-10-22 20:44:09)

# 다음 단계 제안

* Prisma + Supabase 연동으로 일정 CRUD 서버 액션 구현
* React Hook Form + Zod를 접목해 Event Composer 검증/반복 로직 완성
* TanStack Query 캐시 및 Optimistic Update 적용
* 이벤트 상세 패널/수정·삭제 플로우 연결

# 원격 접속 정보 업데이트

**ngrok 터널 URL:** https://baculiform-penny-itchily.ngrok-free.dev

**생성 시각:** 2025-10-22 20:51:06

**터널 프로세스 PID:** 1692623

**주의:** ngrok 무료 플랜은 일정 시간 후 세션이 만료될 수 있으므로, 접속이 끊기면 다시 실행해야 합니다.

# UI 텍스트 조정 기록

**일시:** 2025-10-22 20:53:46

월간 뷰 설명 문구를 제거하고, 상단 배지 텍스트를 "테스트 버젼 1.0V"로 교체했습니다.

# 버전 라벨 업데이트

**일시:** 2025-10-22 20:56:06

주간/월간 뷰 상단 배지를 모두 "테스트 버젼 1.0V"로 통일했습니다.

# 홈페이지 및 회원 기능 확장 계획

마케팅 관점의 홈페이지와 회원 전용 기능 영역을 분리해 신규 입부자 유입과 기존 회원 관리 모두를 충족시키는 구조를 마련한다.

## 1. 홍보용 홈페이지(퍼블릭)

* 히어로 섹션: 합주 활동 사진/영상 배경, "Join LCDA" CTA, 회비 및 모집 일정 요약 배지
* 미션 & 프로그램 소개: 팀 역사, 연례 공연, 워크샵, 대회 수상 경력 소개 카드 UI
* 신입 혜택 & 회비 안내: 입부비, 월 회비, 납부 방법(QR/계좌), 조기 등록 인센티브
* 커리큘럼 타임라인: 학기별 이벤트(오리엔테이션, 리트릿, 정기공연)까지 스크롤 애니메이션
* 지원 플로우: FAQ, 지원 자격, 신청 양식 연결 (`/apply` Next.js form -> Supabase 저장 & Slack Webhook 알림)
* 소셜/뉴스 모듈: Instagram Embed, 공연 영상(YouTube), 언론기사 카드

## 2. 회원 전용 기능 영역

* 인증/인가: Supabase Auth (Email/Password + Magic Link) 또는 Clerk 연동, `role` 필드로 운영진/정회원/준회원 구분
* 접근 제어: Next.js Middleware로 `/calendar`, `/gallery`, `/resources` 등 보호 라우트 적용
* 합주 페이지: 현재 구현한 월간/주간 캘린더 + 일정 생성 권한을 역할별로 제한
* 공지/자료실: 운영진이 올리는 PDF/악보 파일을 Supabase Storage에 저장, 다운로드 로그
* 회비 현황: Stripe Billing Portal 또는 Toss Payments 정기 결제 연동, 납부 현황 대시보드 제공
* 알림: 이메일(Resend) + Kakao 알림톡(추후)을 통해 일정 변경/회비 미납 Remind

## 3. 갤러리 기능 설계

* 콘텐츠 구조: 연도/행사별 앨범 -> 이미지 & 영상 썸네일, Lightbox 뷰어
* 업로드 흐름: 운영진만 업로드 허용, 드래그 앤 드롭 -> Supabase Storage 또는 AWS S3에 저장
* 메타데이터: 제목, 촬영일, 참가 파트, 태그(정기공연/버스킹 등)
* 최적화: 이미지 리사이징(Sharp/Vercel OG Image) 및 WebP 변환, CDN 캐싱
* 앱 내 권한: 정회원은 다운로드 가능, 준회원은 View-only

## 4. 연간 예상 비용(사용자 ~70명)

주요 SaaS/호스팅 조합과 연간 비용 범위를 추정한다. 실제 비용은 사용량에 따라 다를 수 있으며 아래는 안정 운영 기준이다.

**- Vercel Hobby:** 무료 (빌드분 100h/월). 소규모 트래픽 시 무상, 프로플랜 업그레이드 시 $20/월

**- 도메인 + DNS:** Freenom 무료 or .com(약 15,000원/년). Cloudflare DNS 무료

**- Supabase Pro:** $25/월 (스토리지 8GB, DB 8GB). 이미지를 많이 쓸 경우 Storage 추가요금

**- Supabase Auth/Storage 초과:** 추가 스토리지 $0.125/GB-월, 다운로드 $0.09/GB

**- Resend 이메일:** 월 3000건 무료, 초과 시 $20/100k emails

**- Stripe 수수료:** 결제금액의 약 3.3% + 300원 (회비 자동 결제 시 참고)

**- ngrok or Cloudflare Tunnel:** 무료 (테스트용). 운영 시 Vercel 도메인 사용 권장

월평균 약 $25~45 수준 (한화 35~60만원/년)에서 충분히 운영 가능. 스토리지/동영상 트래픽이 급증하면 AWS S3 + CloudFront를 추가 고려해야 함.

업데이트 일시: 2025-10-22 21:02:16

# 현재 데이터 연동 상태 메모

**일시:** 2025-10-22 21:16:47

UI는 일정 생성/수정/삭제 플로우를 위한 컴포넌트를 준비했지만, 아직 서버 액션과 데이터베이스 연동은 연결되지 않은 상태이다.

**무료/저비용 데이터베이스 옵션:**

* Supabase Free Tier: Postgres 500MB, Auth/Storage 포함. 초기 단계 무료, Pro로 확장 시 $25/월.
* Neon Free Plan: Serverless Postgres 3GB/일, 1GB 스토리지. Prisma와 연동 용이.
* Railway Free: Postgres 인스턴스 월 $5 크레딧 내 무료. 자동 슬립 포함.
* PlanetScale Hobby: MySQL 호환, 5GB 저장공간 무료 (단, 외부 스토리지가 필요).
* 로컬 개발: SQLite 파일(Prisma) or Docker Postgres → 운영 이전.

**다음 단계 제안:**

1. 1) Supabase Free로 초기 연결 → Prisma schema 작성 후 서버 액션 구축
2. 2) 일정 생성/수정/삭제 API → TanStack Query로 프론트와 동기화
3. 3) 필요 시 비용/엔터프라이즈 전환 전에 데이터 마이그레이션 전략 수립

# 데이터베이스 및 로컬 저장 구성 업데이트

**일시:** 2025-10-22 21:59:27

* Prisma ORM 도입, SQLite 로컬 DB(`prisma/dev.db`)로 일정 저장 구조 구성 – 추후 Postgres(Neon/Supabase)로 마이그레이션 가능하도록 모델 설계
* Event 테이블: title, description, location, color, start, end, timezone, recurrenceRule(JSON 문자열) 등 필드 정의
* Next.js App Router API(`/api/events`)에 CRUD 라우트 추가 – Zod 검증 + 반복 일정 RRULE 처리
* TanStack Query + React Hook Form + Prisma 연동으로 일정 생성/수정/삭제 UI 연결, 반복 옵션까지 반영
* 로컬 dev 서버 PID 및 ngrok 공유 링크는 필요 시 수동 종료/재시작; EADDRINUSE 방지를 위해 단일 프로세스 유지 필요

## 향후 마이그레이션 메모

* DATABASE\_URL 교체 시 Prisma `schema.prisma` provider를 postgresql로 변경, `prisma migrate deploy`로 이관
* Supabase/Neon 사용 시 커넥션 풀링(PgBouncer) 고려, Prisma `pgbouncer` 설정 적용
* 반복 일정 예외(EventException) 테이블 확장, 스트리밍 Query/Realtime 연동은 차후 단계

# UI 상세 조정 기록

**일시:** 2025-10-23 02:18:53

* 헤더가 일정 모달/상세 시트 오버레이 시 자동으로 위로 슬라이드되어 가려지지 않도록 overlay 상태 플래그를 도입
* 월간/주간 캘린더 섹션과 오버뷰 카드 폭을 1100px 기준으로 맞추어 중앙 정렬
* 이벤트 상세 시트를 오른쪽 여백 대신 화면 중앙에 가까운 위치로 이동하고 높이·버튼 정렬 개선
* 월간 뷰에서 마우스를 올린 날짜에만 + 버튼이 나타나도록 상태 기반 hover 로직 추가
* 오늘 날짜 배지를 붉은 원 배경에 검은 텍스트로 변경해 가독성을 향상

# UI 상세 조정 기록

**일시:** 2025-10-23 02:28:12

* 헤더가 일정 모달/상세 시트 오버레이 시 자동으로 위로 슬라이드되어 가려지지 않도록 overlay 상태 플래그를 도입
* 월간/주간 캘린더 섹션과 오버뷰 카드 폭을 1100px 기준으로 맞추어 중앙 정렬
* 이벤트 상세 시트를 오른쪽 여백 대신 화면 중앙에 가까운 위치로 이동하고 높이·버튼 정렬 개선
* 월간 뷰에서 마우스를 올린 날짜에만 + 버튼이 나타나도록 상태 기반 hover 로직 추가
* 오늘 날짜 배지를 붉은 원 배경에 검은 텍스트로 변경해 가독성을 향상

# UI 상세 조정 기록

**일시:** 2025-10-23 02:42:55

* 이벤트 상세 UI를 오른쪽 패널 대신 모달 형태로 중앙에 배치하고, 날짜별 목록 보기 ↔ 일정 상세 전환 구조 구현
* 여러 일정이 겹칠 때 “더보기”를 누르면 해당 날짜의 모든 일정이 모달 내 리스트로 표시되고, 개별 선택 시에만 수정/삭제 버튼 노출
* 일정 생성 모달에 달력 팝오버 기반 날짜 선택 UI 도입(월 이동, 오늘 강조, 선택 강조 포함)
* 월간 뷰의 오늘 배지를 빨간 배경/검은 텍스트로 재디자인하고, Hover된 날짜에만 + 버튼이 나타나도록 개선
* 상세 모달/작성 모달 오버레이 시 헤더가 자동으로 사라져 콘텐츠가 가려지지 않도록 처리

# 주간 타임라인 로직 조정

**일시:** 2025-10-23 03:04:28

* 주간 뷰 데이터 필터링을 `event.start <= rangeEnd && event.end >= rangeStart` 방식으로 수정해 모든 겹치는 일정이 포함되도록 함
* WeekTimeline에서 하루 단위 필터를 시작/종료 시각으로 클램프하여, 한 날짜에 여러 일정이 있을 때 정확한 시간대에 표시되도록 조정
* 이슈 재현 시 저장된 DB의 일정이 주간 뷰에도 동일하게 반영됨을 확인 (테스트 없이 로직만 반영)

# 주간 타임라인 UI 보정

**일시:** 2025-10-23 03:10:25

* 주간 뷰 이벤트 배치를 30분 단위 슬롯 계산에서 분 단위 비율 계산으로 전환하여, 시간대가 정확히 배치되도록 조정
* 표시 높이/위치를 클램프해 하루 범위를 초과하는 일정도 올바른 영역만 채우도록 처리
* 하루 이벤트 목록을 시작 시간 기준으로 정렬해 겹치는 일정이 순차적으로 표시되도록 함

# 뷰 전환 포커스 개선

**일시:** 2025-10-23 03:16:08

* 월간 뷰에서 일정/일자를 클릭하거나 더보기로 진입하면 anchorDate를 해당 날짜로 갱신해 주간 뷰 전환 시 동일한 주가 표시되도록 개선
* 주간 뷰에서도 일정 클릭 시 anchorDate를 업데이트하여 상세 모달이 열린 상태에서도 주간 타임라인이 맞는 날짜를 보여줌

# 주간 타임라인 슬롯 재계산

**일시:** 2025-10-23 03:24:17

* 주간 뷰 이벤트 배치를 30분 슬롯 인덱스 기반으로 재구성해, 30분 단위에 정확히 맞춰 위치/높이를 계산
* 시작/종료가 하루 경계를 넘는 경우에도 슬롯 단위로 클램프하여 렌더링 위치가 틀어지지 않도록 수정

# 주간 타임라인 타임존 보정

**일시:** 2025-10-23 03:31:51

* 주간 뷰 이벤트 위치 계산 시 `date-fns-tz`를 활용해 각 일정의 timezone(예: Asia/Seoul)에 맞춰 슬롯 위치를 계산
* 일정 시작/종료를 분 단위로 클램프해 시간대가 다른 사용자 환경(UTC 등)에서도 정확한 위치·높이로 렌더링
* 30분 슬롯 기준의 최소 높이를 유지해 겹치는 일정도 안정적으로 표시

# 주간 타임라인 슬롯 정렬 재수정

**일시:** 2025-10-23 03:37:02

* 시간대가 어긋나던 주간 뷰 이벤트를 30분 슬롯 인덱스로 다시 계산해 오전 일정이 하단으로 밀리지 않도록 수정
* 시작/종료는 하루 범위로 클램프하고, 최소 30분 높이를 확보해 파란 막대가 정확한 박스 영역에 들어가도록 처리