

현장 서비스 관리(FSM) · B2B SaaS · 풀스택 구축

# 현장 기술자 파견 스케줄링 B2B SaaS 플랫폼

— 미국·캐나다 500명+ 테크니션을 프로젝트에 배정하고, 변경을 이메일·SMS로 즉시·정확하게  
통지하는 단일 스케줄링 도구 —

드래그앤드롭 캘린더

ScheduleEvent 이벤트 소싱

실시간 동기화

AWS SES · SNS 알림

도메인 현장 서비스 관리(FSM) · 인력 배정 SaaS

담당 역할 풀스택 단독 (설계 · DB · BE · FE · 인프라)

개발 기간 8주 (63 역일 · 업무 56 + 검수 7)

규모 14 도메인 모델 · 17 API 라우트 · 테스트 276건

MVP 개발 완료

Phase 4 UAT 진행

TABLE OF CONTENTS



목차

<b>1</b>	<b>프로젝트 개요</b> .....	<b>03</b>
<b>2</b>	<b>주요 기능 ① — 스케줄러 · 테크니션</b> .....	<b>04</b>
<b>3</b>	<b>주요 기능 ② — 알림 · 인증 · 운영 · 실시간</b> .....	<b>05</b>
<b>4</b>	<b>프로젝트 배경 — 문제 · 목표 · 해결</b> .....	<b>06</b>
<b>5</b>	<b>기술 스택 · 아키텍처</b> .....	<b>07</b>
<b>6</b>	<b>핵심 기능 상세</b> .....	<b>08</b>
<b>7</b>	<b>진행 단계</b> .....	<b>09</b>
<b>8</b>	<b>프로젝트 성과</b> .....	<b>10</b>
<b>9</b>	<b>라이브 데모 · 유사 경험</b> .....	<b>11</b>

OVERVIEW

# 1 프로젝트 개요

미국·캐나다 전역에 500명 이상의 현장 기술자(테크니션)를 파견하는 사업자가, 스케줄러는 프로젝트 별 인력 배정을 드래그앤드롭으로 관리하고 그 변경을 테크니션이 이메일·SMS로 즉시·정확하게 확인 하도록 하는 B2B SaaS 플랫폼을 신규 구축한 프로젝트입니다.

**500+**

운영 테크니션

**14**

DB 도메인 모델

**17**

API 라우트 모듈

## ■ 작업 범위



### 개발 범위

제품 설계 · 데이터 모델 · 백엔드 API · 프론트엔드 · 알림 · 인프라 · 배포까지 풀스택 단독 수행.



### 지원 환경

스케줄러·관리자는 PC Web 전용 (Chrome/Safari/Edge), 테크니션은 모바일 브라우저(앱 설치 불필요).

## ■ 주안점

- **프로젝트 할당 모델** — 셀 복사가 아닌 "프로젝트를 하나의 객체로", 이동 시 배정 인원이 함께 이동
- **이벤트 소싱 정확도** — 모든 변경을 ScheduleEvent로 기록해 화면 = DB 일치를 구조적으로 보장
- **알림 신뢰성** — 재시도 · 채널 에스컬레이션 · 발송 증빙으로 전화 통지 회귀 차단
- **로컬 = AWS 1:1 미러** — Docker 컨테이너가 AWS 리소스와 대응, 환경변수 교체만으로 prod 전환

FEATURES — SCHEDULER & TECHNICIAN

## 2 주요 기능 ① — 스케줄러 · 테크니션

### ■ 스케줄러 기능 (PC)

**📅**

#### 드래그앤드롭 캘린더

테크니션 x 일자 그리드에서 프로젝트 막대를 드래그(Full Bar Delta Shift), react-window 가 상 스크롤.

**🔄**

#### 프로젝트 일괄 이동

프로젝트를 옮기면 소속 배정 (Assignment) 전체가 같은 delta만큼 함께 이동.

**✓**

#### 발송 증빙 토스트

이동 직후 "3/3 테크니션 통지 완료"를 표시 — 전화 통지를 대체 하는 핵심 UX.

### ■ 테크니션 기능 (모바일)

**📱**

#### 오늘의 스케줄 카드

프로젝트명 · 주소 · 시간 · 장비를 하단 탭 바 구조로 한눈에.

**📄**

#### 변경 알림함

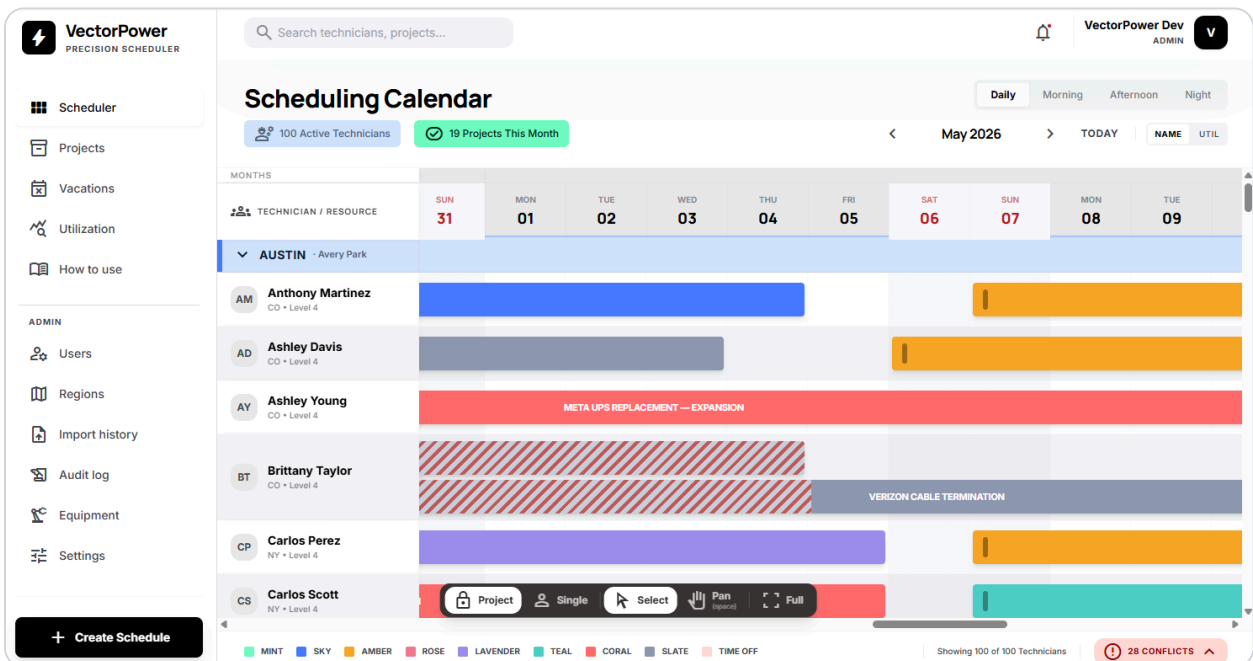
변경 내역(before/after) 확인 + 본인 읽음 상태(readAt) 관리.

**📅**

#### 휴가 신청

향후 7일 일정 + 휴가 신청-이력을 모바일에서 직접 처리.

### ■ 대표 화면 — 스케줄러 캘린더






좌측 테크니션 명단·가동률 사이드바 + 우측 일자별 프로젝트 색상 막대 + 상단 월/팀 네비게이션

FEATURES — NOTIFY · AUTH · OPS

### 3 주요 기능 ② — 알림 · 인증 · 운영 · 실시간

#### ■ 알림 · 인증 · 실시간

 <p><b>이메일 · SMS 알림</b></p> <p>AWS SES + SNS 동시 발송, BullMQ 큐 워커가 처리. 실패 시 최대 3회 재시도 후 대체 채널에 스칼레이션.</p>	 <p><b>매직 링크 인증</b></p> <p>무비밀번호 로그인(1회용 토큰 15분·SHA-256 해시), JWT Access+Refresh, Redis rate limit.</p>	 <p><b>실시간 동기화</b></p> <p>Socket.io 브로드캐스트 + 배경 version 기반 Optimistic Locking으로 동시 편집 충돌 처리.</p>
--	---	---

#### ■ 운영 · 관리

 <p><b>휴가 신청·승인</b></p> <p>신청 → 스케줄러/관리자 승인·반려, 배정 충돌 경고, 결정 알림.</p>	 <p><b>가동률 대시보드</b></p> <p>테크니션별 현재 프로젝트·가동률·배정/가용일수를 밀집 테이블로.</p>	 <p><b>CSV 일괄 등록</b></p> <p>사용자·프로젝트 CSV 업로드, 행 단위 검증·오류 리포트.</p>
 <p><b>프로젝트 · 임시 이탈</b></p> <p>프로젝트 CRUD·상세 패널, 8색 색상, Detachment 이탈/복귀 관리.</p>	 <p><b>감사 로그</b></p> <p>폴리모픽 entity 변경 이력 + 필터·상세 모달.</p>	 <p><b>리전 · 시스템 설정</b></p> <p>스케줄러-지역 그룹핑, 가동률 임계값 등 런타임 설정.</p>

BACKGROUND

## 4 프로젝트 배경

### ■ 문제점

- **연쇄 수동 조정** — 스케줄러 1인이 70~80명 관리, 프로젝트 이동 시 배정 인원을 일일이 손으로 재배치
- **이중 도구** — Google Sheets는 스케줄러용이나 테크니션에게 안 가고, Teams Shifts는 테크니션용이나 스케줄러 인터페이스로 부적합
- **전화 통지** — 변경마다 테크니션에게 전화 → 시간 소모 극심, 확장 불가
- **정보 부정확 = 실제 비용** — 잘못된 스케줄은 기술자의 잘못된 이동(운전/비행)·고객 불만으로 직결

### ■ 목표 → 해결 방안



### ■ 핵심 해결 포인트

<div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="background-color: #e6f2ff; border-radius: 50%; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center; margin-right: 5px;"> <span style="font-size: 12px;">뭉</span> </div> <div> <p><b>프로젝트 할당 모델</b></p> <p>"1명→1잡→1일" 디스패치 모델이 아닌 "팀→다주차 프로젝트→중간 이탈/복귀" 모델. 기존 FSM과의 근본적 차별점.</p> </div> </div>	<div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="background-color: #e6f2ff; border-radius: 50%; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center; margin-right: 5px;"> <span style="font-size: 12px;">✓</span> </div> <div> <p><b>이벤트 소싱 정확도</b></p> <p>모든 변경을 ScheduleEvent로 적재하고 최신 이벤트 체인으로 렌더링 → 화면=DB 일치를 코드 레벨에서 보장.</p> </div> </div>
---	---

TECH STACK & ARCHITECTURE

# 5 기술 스택 · 아키텍처

## FRONTEND

- Next.js 14
- React 18
- TypeScript
- Tailwind CSS
- dnd-kit
- react-window
- TanStack Query
- Zustand

## BACKEND

- Node.js 20
- Express 4
- Socket.io
- BullMQ
- Redis 7
- JWT
- Zod
- helmet

## SHARED · DB

- PostgreSQL 16
- Prisma ORM
- citext · uuid v7
- @vp/shared (Zod 14종)
- npm 모노레포

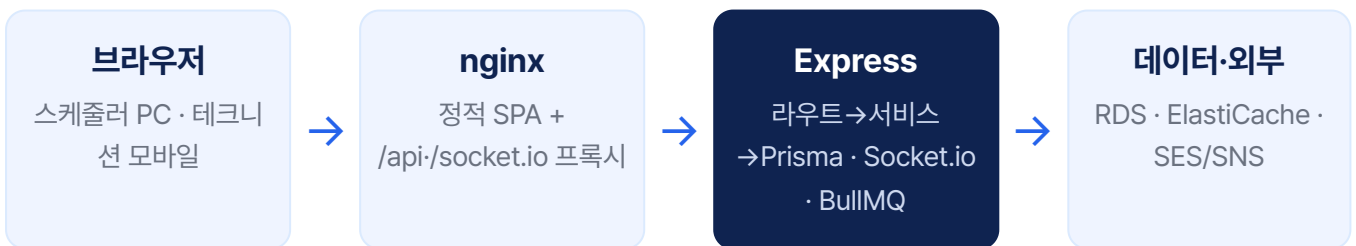
## EMAIL / SMS

- AWS SES
- AWS SNS
- Mailpit (dev)

## INFRA

- AWS EC2 · RDS · ElastiCache
- Docker Compose
- nginx
- GitHub Actions

### ■ 배포 아키텍처 (단일 EC2 + 2 컨테이너)



\* Docker Compose 로컬 컨테이너가 AWS 리소스와 1:1 미러 — prod 전환 시 환경변수만 교체.

## KEY FEATURES IN DEPTH

## 6 핵심 기능 상세

### 1 드래그앤드롭 캘린더 (Full Bar Delta Shift)

dnd-kit DndContext를 react-window 가상 스크롤 외부에 배치, 셀 ID 규약으로 연결. 프로젝트 막대 드래그 시 시작·종료일과 모든 배정이 같은 delta로 시프트. 15,000셀 60fps를 1주차 spike로 선검증해 모달 Plan B를 불필요화.

### 2 ScheduleEvent 이벤트 소싱 + 캐시 컬럼

변경 시 트랜잭션 안에서 배정 캐시 컬럼(currentStart/End/Status)을 갱신하고 before/after JSON 이벤트 적재. 캘린더 조회는 캐시 컬럼 인덱스로 단순·고속 처리.

### 3 알림 큐 + 재시도 + 채널 에스컬레이션

BullMQ 워커가 SES·SNS 발송, 실패 시 동일 채널 3회 재시도 후 대체 채널로 전환. SES 바운스/컴플레인 webhook 수신 시 이메일 자동 억제(emailSuppressedAt).

### 4 매직 링크 인증 + Rate Limit

1회용 토큰 SHA-256 해시 저장(15분), SES 발송 → Access(1h)+Refresh(7d) 발급. 요청·검증에 Redis sliding-window rate limit으로 남용·enumeration 방어.

### 5 실시간 동기화 + 동시 편집 충돌

Socket.io로 변경을 타 스케줄러 화면에 전파. 배정 version 기반 Optimistic Locking으로 늦은 저장에 충돌 토스트 + 자동 리프레시.

### 6 CSV 일괄 등록

multer + csv-parse로 사용자·프로젝트 업로드, 행 단위 검증·적재. 실패 행은 사유와 함께 ImportBatch에 기록·조회.

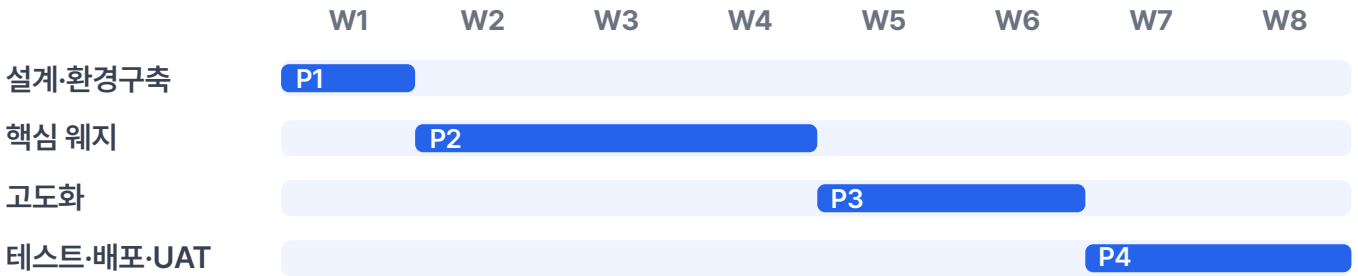
### 7 로컬 = AWS 미러 인프라

dev 4컨테이너(postgres·redis·mailpit·backend)가 AWS와 대응. prod는 단일 EC2에 backend + nginx 2컨테이너, 외부 진입은 nginx만 노출(backend expose).

PROCESS

# 7 진행 단계

총 8주(63 역일) · 웨지 우선(Wedge-First) — 핵심 루프를 2~4주차로 당겨 4주차에 양측 실사용 MVP 완성



## ○ P1 설계 · 환경 구축 (1주차)

- 데이터 모델(14개 도메인 모델·10개 enum)·API·테스트 설계서 작성
- Docker Compose 로컬 환경 구성 — postgres·redis·mailpit·backend 컨테이너를 AWS 리소스와 1:1 미러
- AWS SES 프로덕션 승인 즉시 신청 (리드타임 대비 Phase 2 알림 작업 차단 방지)
- 스케줄러 실제 작업 화면 공유·녹화 요청으로 관찰 간극 보완

## ○ P2 핵심 웨지 (2~4주차)

- 매직 링크 무비밀번호 인증 + JWT(Access·Refresh) 세션 + Redis rate limit
- 드래그앤드롭 캘린더(dnd-kit + react-window 가상 스크롤) + 프로젝트 단위 일괄 이동
- ScheduleEvent 이벤트 소싱 + 테크니션 모바일 뷰(오늘의 카드·변경 알림함)
- 이메일·SMS 알림 발송 + 발송 증빙 토스트 → 4주차 종료 시 양측 실사용 MVP 완성

## ○ P3 고도화 (5~6주차)

- 프로젝트 CRUD·슬라이드오버 상세 패널, 임시 이탈/복귀(Detachment) 관리
- 휴가 신청·승인 워크플로우, 가동률 대시보드(운영진)
- Socket.io 실시간 동기화 + 배경 version 기반 Optimistic Locking 충돌 처리
- 리전·장비·감사 로그·CSV 일괄 등록·시스템 설정 등 관리자 영역

## ○ P4 테스트 · 배포 · UAT (7~8주차)

- 회귀 테스트(약 276건) + DB 제약·트랜잭션·동시성 통합 테스트
- AWS prod 배포 — 단일 EC2에 backend + nginx 2컨테이너, RDS·ElastiCache 연결
- 보안 점검(helmet·nginx 헤더·OWASP) + 부하 테스트로 인덱스 전략 검증
- UAT 피드백 hotfix 사이클 (P0 1시간·P1 4시간 SLA)

RESULTS

## 8 프로젝트 성과

### ■ 정량 성과 (코드 / 규모 기준)



### ■ 정성 성과

- **단일 루프 완성** — 스케줄러 변경부터 테크니션 확인까지를 하나의 도구로 통합, 이중·삼중 작업 해소
- **구조적 정확도 보장** — 이벤트 소싱 + 캐시 컬럼으로 "화면 = DB" 일치를 코드 레벨에서 보장
- **알림 신뢰성 확보** — 재시도·채널 에스컬레이션·발송 증빙으로 전화 통지 회귀 차단
- **dev/prod 패리티** — 로컬 = AWS 1:1 미러로 배포 리스크 사전 제거

### ■ 품질 · 운영 지표



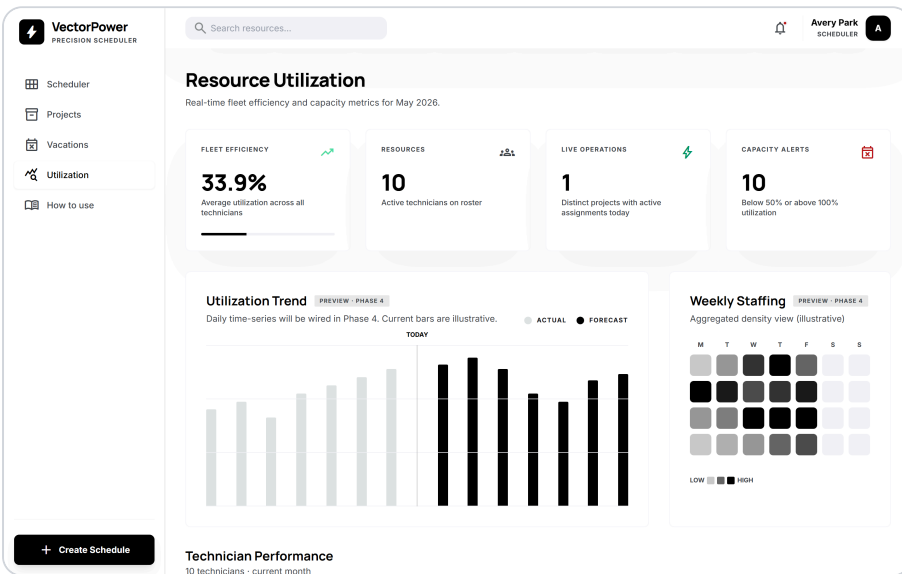
LIVE DEMO & EXPERIENCE

# 9 라이브 데모 · 유사 경험

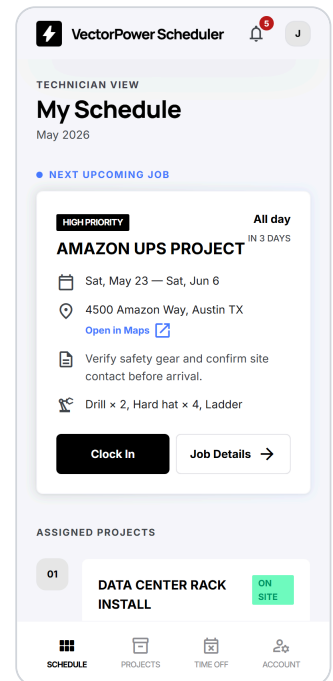
## ■ 라이브 데모 · 접속 구조

- 운영 도메인 — 클라이언트 내부 운영 도구로 비공개 (계약 후 데모 계정 별도 안내)
- 스케줄러 · 관리자 — PC 브라우저(Chrome/Safari/Edge) 전용
- 테크니션 — 모바일 브라우저로 접속, 앱 설치 불필요
- 배포 — AWS EC2(HTTPS) + nginx, GitHub Actions CI/CD

## ■ 대표 화면



가동률 대시보드 (운영진) — 효율·리소스·용량 경보 KPI + 가동률 추이·주간 배치



테크니션 모바일 뷰

## ■ 유사 프로젝트 경험

**현장 서비스 리포트 시스템**

현장 인력 대상 모바일 우선 UX, 오프라인 동기화, PDF 자동 생성, AWS 인프라 구축.

**LMS · B2B SaaS 구축**

AWS ECS/EC2, JWT 인증, RBAC, 관리자 대시보드, 모노레포 기반 풀스택 다수.