

# Potfolio

## : 시작점

**강나현**, FrontEnd Developer

비전공자라는 배경은 한계가 아닌, 압축된 성장의 동력이었습니다.

짧은 시간 안에 이뤄낸 이 결과물은 저의 열정과 노력의 첫 결실이며,

앞으로 더 큰 성장을 이뤄낼 소중한 **시작점**이라고 믿습니다.

**Language**

Javascript, Typescript, HTML/CSS, JAVA, Python

**FrontEnd**

React, Next.js

**Tools**

Git, Figma, Ngrok

## 프로젝트 환경

언어 Typescript, HTML/CSS, JAVA, Python

프레임워크 React, Next.js, Spring Boot

DB Mysql

API REST API, FastAPI

Tools Git, Figma, Google Docs

---

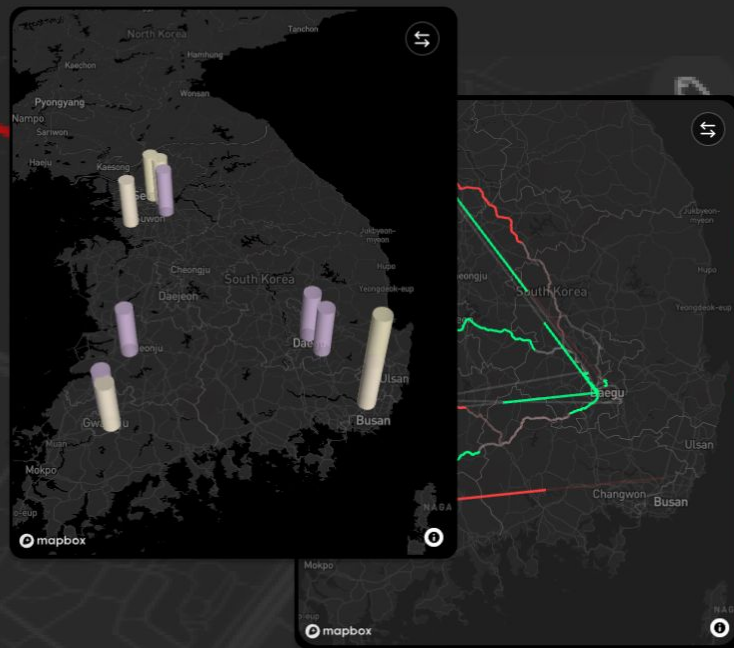
기간 2025.06. ~ 2025.07 (5주)

플랫폼 Web

개발 인원 3명 (FrontEnd 1명, BackEnd 1명, DA 1명)

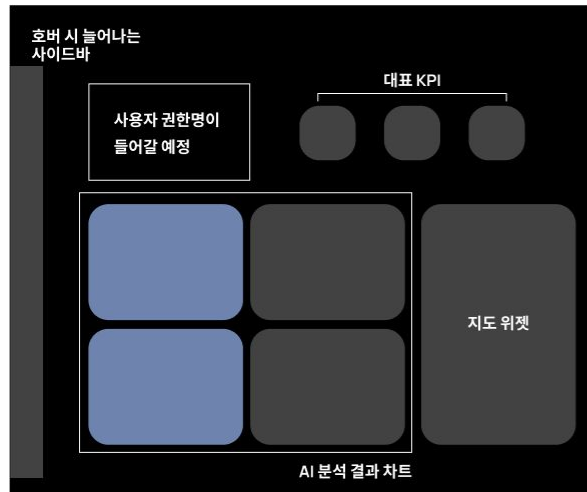
담당 역할 FrontEnd

## AI 지능형 EPC 기반 유통 흐름 시각화 : 불법 유통 추적 및 알림 시스템



# 데이터 인사이트를 위한 Key 설계 목표

## 대시보드



### 직관적인 대시보드 레이아웃

- 한눈에 AI 분석 결과를 보여주기 위하여 Statcard 및 위젯 사용
- 호버 시에만 사이드바가 늘어나도록 하여 화면 공간의 효율적 사용
- 프로젝트 주요 기업의 로고 색 활용

## 시각화

라이브러리	장점	단점
D3.js	<ul style="list-style-type: none"><li>• 데이터의 enter/update/exit를 세밀하게 제어할 수 있음</li><li>• 웹 표준(SVG/HTML)을 이용한 범용적이고 창의적인 데이터 시각화</li></ul>	<ul style="list-style-type: none"><li>• 수천 개 이상의 SVG 요소를 렌더링하면 급격히 느려짐</li><li>• 지도 이미지 직접 구현해야 함</li><li>• 배경 움직이면 노드 위치 이상해짐</li></ul>
React Flow	<ul style="list-style-type: none"><li>• 커스텀 혹은 통해 노드와 엣지의 상태를 매우 직관적으로 관리</li></ul>	<ul style="list-style-type: none"><li>• 복잡한 커스터마이징의 어려움</li><li>• 3D 시각화를 지원하지 않음</li></ul>
Deck.gl	<ul style="list-style-type: none"><li>• 수백만 개의 점/선을 부드럽게 렌더링</li><li>• 지리 시각화에 최적화된 레이아웃 제공</li><li>• 손쉬운 3D 시점, 투영, 카메라 제어</li></ul>	<ul style="list-style-type: none"><li>• WebGL을 지원하지 않는 환경에서는 사용 불가</li></ul>

### 3차원으로 유동 상황을 시각화할 라이브러리 선정

- \*3차원 시각화가 필수적이고 대용량 데이터를 다루어야 하는 프로젝트 특성상 가장 적합하다 판단하여 최종적으로 Deck.gl 선정
- Deck.gl + MapBox 조합으로 커스텀 지도 활용

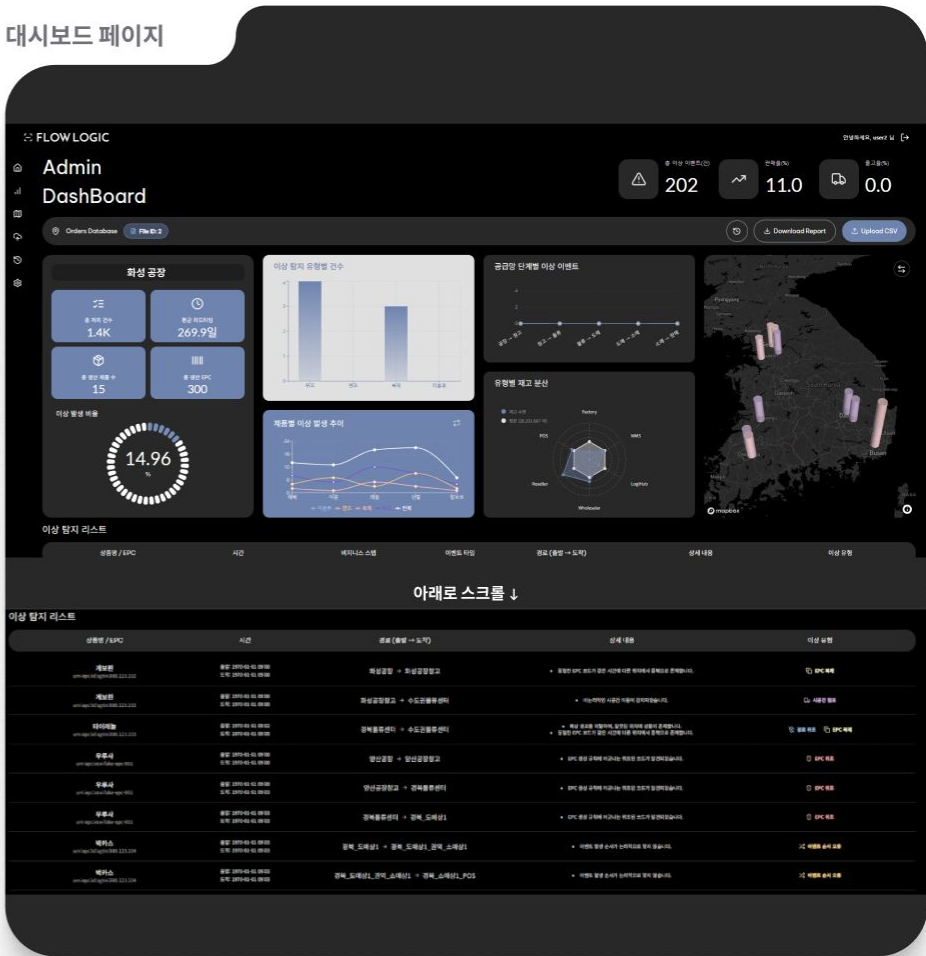
\*3차원 : 시간, 공간, 이벤트 타입(입고, 출고 등)을 의미함.

## AI 분석 보고서



### 템플릿 기반 리포트

- 대시보드를 구성하는 컴포넌트(차트, 표, 지도 등)와 보고서의 구성 요소를 동일한 코드 사용



# "한 점에서 시작하여, 전체 흐름을 꿰뚫다."

## Jotai 아톰으로 엮어낸, 끊임 없는 데이터 추적 경험

수정 전

useEffect 기반 데이터 관리

useEffect의 의존성 배열이 복잡해졌고,  
하나의 상태 변경이 다른 useEffect를 촉발,  
그것이 또 다른 상태를 변경하는 "무한 루프" 발생

수정 후

아톰 기반 데이터 관리

아톰을 구독하기만 하면  
항상 최신의,  
그리고 동일한 상태를 공유하게 됨

useEffect 기반의 지역적 상태 관리에서 Jotai 아톰과 사용자 정의 훅을 활용한 전역적, 원자적 상태 관리로 전환한 것은 이 프로젝트의 중요한 전환점이었습니다.

사용자가 히트맵에서 특정 항목을 클릭하면, 그 선택이 즉시 흐름도로 전달되어 해당 경로가 포커싱되는 유기적인 분석이 가능합니다.

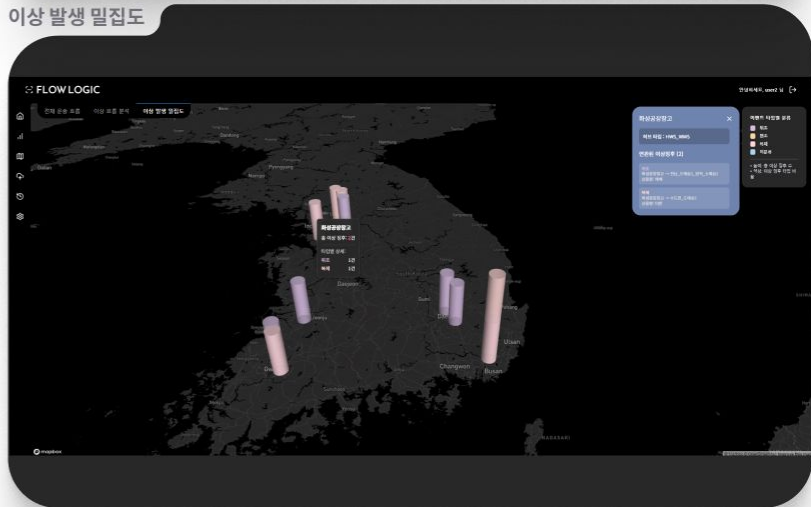
이 모든 뷰의 상태는 Jotai의 아톰으로 중앙 관리되기 때문에, 탭을 전환하더라도 데이터는 물론 카메라 위치와 같은 UI 컨텍스트까지 완벽한 무결성을 유지합니다.

그 결과, 사용자는 기술적인 제약 없이 오직 데이터 분석의 흐름에만 집중할 수 있습니다.

### 이상 흐름 분석



### 이상 발생 밀집도



# "AI 보고서 생성 기능 심층 구현기"

## PDF 렌더링의 숨겨진 과제들

- ❌ **문제 1: Tailwind CSS 해석 불가로 인한 렌더링 및 다운로드 실패**

PDF 다운로드 기능을 위해 사용한 `html2canvas`는 빌드 시점에 동적으로 생성되는 Tailwind의 유틸리티 클래스를 전혀 해석하지 못해 계속해서 다운로드를 실패하였습니다.

✅ **해결: 인라인 스타일 변환**

`style` 속성을 이용한 인라인 스타일을 적용하고 구체적인 색상 코드로 변환하여 해결하였습니다.
- ❌ **문제 2: 콘텐츠 잘림 및 페이지네이션 부재**

스크롤이 필요한 긴 데이터 테이블은 A4 용지 크기를 넘어가면서 그대로 잘려버렸고, 자동 페이지 나누기나 쪽수 표시 기능이 없어 전문적인 보고서 형태를 갖추기 어려움을 겪었습니다.

✅ **해결: 하이브리드 접근 방식의 도입**

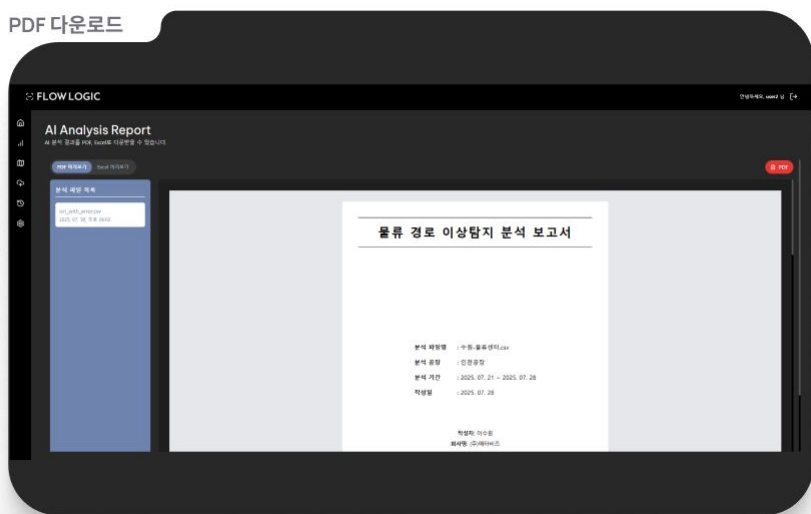
시각적으로 복잡한 차트는 `html2canvas`로 캡처하고, 양이 많은 상세 데이터 테이블은 `jspdf-autotable` 라이브러리를 사용해 PDF에 동적으로 직접 그리도록 했습니다. 이 라이브러리는 자동 페이지 나누기와 쪽수 표시를 완벽하게 지원합니다.

간단할 것이라 생각했던 PDF 생성 기능은, 디테일한 문제 해결 과정의 연속이었습니다.

문제 해결을 위해 대한통운과 같은 업계 선두 기업의 보고서 형식을 참고하며 사용자가 기대하는 표준을 파악하고, 이를 기술적으로 구현하기 위해 하이브리드 아키텍처를 고안했습니다.

이 과정은 제게 단순한 기능 개발을 넘어, 문제를 정의하며 끝내 해결해내는 값진 성장을 안겨주었습니다.

## PDF 다운로드



## Excel 다운로드



## 좋았던 점

실제 물류 기업의 데이터를 활용하여 프로젝트를 진행할 수 있었던 점이 가장 좋았습니다. 정제된 더미 데이터가 아닌, 현장의 복잡성과 특수성이 담긴 데이터를 다루면서 현실적인 문제 해결 능력을 기를 수 있었습니다.

또한, 프로젝트 초기부터 꿈꿔왔던 **3D 데이터 시각화**를 구현하는 과정은 그 자체로 큰 도전이자 배움의 기회였습니다.

특히, 복잡한 지리 정보 위에 3D 데이터를 렌더링하고 유의미한 인터랙션을 만들어내는 과정은 쉽지 않았지만, 그 어려움을 극복하며 Deck.gl과 같은 전문 라이브러리에 대한 깊이 있는 이해와 자신감을 얻게 되었습니다.

이 프로젝트를 통해 현실의 데이터를 다루는 감각과, 추상적인 아이디어를 구체적인 기술로 실현해내는 값진 경험을 얻을 수 있었습니다.





# 아쉬운 점

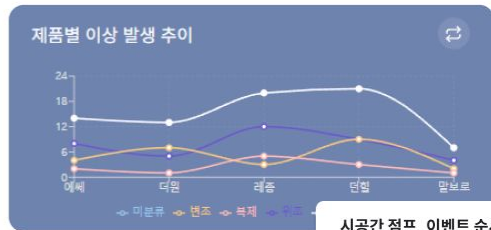
프로젝트를 진행하며 가장 아쉬웠던 점은, 예상치 못한 변수들로 인해 UI/UX 디테일을 더욱 깊이 있게 다듬을 시간을 충분히 확보하지 못했던 것입니다.

초기에는 프론트엔드 개발이 2인으로 기획되었으나, 중간에 1인으로 진행하게 되면서 모든 역할을 혼자 감당해야 했습니다.

또한, 프로젝트 마감 직전에 데이터 분석팀으로부터 기존의 5가지 이상 탐지 라벨이 아닌, 단순 이상 여부만 전달받게 되는 갑작스러운 변경 사항이 있었습니다. 이로 인해 백엔드 개발자와 함께 새로운 룰베이스 로직을 급하게 수립하고, 최종 4개의 라벨에 맞춰 프론트엔드 코드를 수정하는 데 예상보다 많은 시간을 할애해야 했습니다.

물론, 초기 설계 시 types 파일로 정리하여 라벨링 관련 코드를 한곳에서 관리하도록 추상화해두었기에, 실제 코드 수정은 빠르게 대처할 수 있었습니다.

하지만 이 과정에서 UI/UX의 완성도를 한 단계 더 끌어올릴 수 있었던 소중한 시간을 사용하게 된 점은 아쉬움으로 남습니다.



시공간 점프, 이벤트 순서 오류, 위조, 복제, 경로 위조로 총 5가지 유형에서

위조, 변조, 복제, 신규 유형 총 4가지로 최종 변경됨



```
▼ types
TS anomalyUtils.ts
TS api.ts
TS chart.ts
TS colorUtils.ts
TS data.ts
TS datareal.ts
TS error.ts
TS file.ts
TS join.ts
TS map.ts
```



# 도전하고 싶은 점

이번 프로젝트는 데이터를 효과적으로 '제공'하는 데에 초점을 맞추었습니다.  
이제 이 기반 위에서, 사용자가 더욱 능동적으로 분석에 참여하고 깊이 있는 인사이트를 얻을 수 있도록 다음과 같은 기능들을 보완하고 싶습니다.

## • 전문가 의견 주석 기능 :

- 도전 목표: 현재는 이상 탐지 이벤트를 보여주지만 하지만, 앞으로는 각 이벤트에 대해 사용자가 직접 '왜 이것이 이상이라고 생각하는지', '어떤 조치가 필요했는지' 등의 전문가적 의견이나 메모를 남길 수 있는 인터랙션을 추가하고 싶습니다. 이는 단순한 데이터 조회를 넘어, 시스템과 전문가의 지식이 함께 축적되어 **AI 학습에 다시 활용하는 분석 플랫폼**으로 발전하는 첫걸음이 될 것입니다.

## • 룰베이스 vs AI 분석 비교 패널 도입 :

- 도전 목표: 현재의 '이상 탐지 리스트'를 더욱 발전시켜, '룰베이스 기반 이상 이벤트'와 'AI 모델 기반 이상 이벤트'를 나란히 보여주는 두 개의 패널로 확장하고 싶습니다. 이는 두 가지 탐지 방식에 우열을 두는 것이 아니라, 사용자가 두 결과를 비교 분석하며 "**AI는 예측하지 못했지만 룰에 걸린 이벤트**" 또는 **그 반대의 경우**를 탐색하며 더 복잡적이고 정교한 인사이트를 발견할 수 있도록 돕는 기능입니다.

이러한 도전 과제들을 통해, 저는 단방향 정보 제공을 넘어 사용자와 시스템이 상호작용하며 가치를 창출하는 고도화된 대시보드를 만들어나가는 경험을 하고 싶습니다.



상세 정보 패널에서 의견을 남길 수 있도록 함

복제품 의심 이력

선택된 경로와 동일한 EPC를 사용하는 복제품 의심 경로 목록입니다.

더원

화성공정참고 >> 수도권\_도매상1

더원

전남물류센터 >> 영산공장

선택된 경로 상세

복제

기존 EPC를 그대로 복사

상품명 : 더원

EPC : 1.880.123.562

LOT ID : LOTB-2

Event Type : WMS\_Outbound

도착지

수도권\_도매상1

1970-01-01-09:01:06

출발지

화성공정참고

1970-01-01-09:00:16

감사합니다.



**Git Hub**

<https://github.com/nang5587/ai-anomaly-monitor>