

PROJECT

BLUECOOL

CMS & Blog Platform Architect

Developer I Min Pyo
pmini1203@gmail.com

2026.01

01



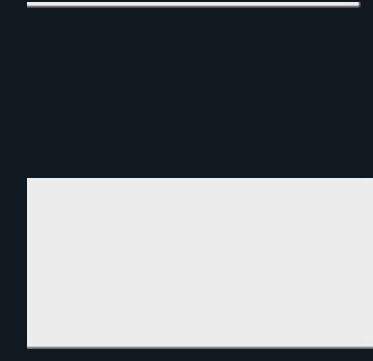
Service Overview

02



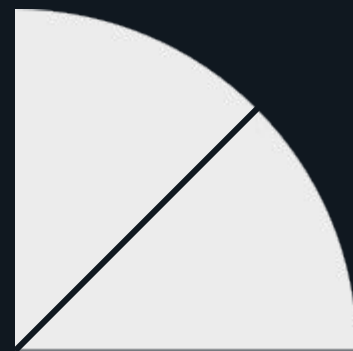
Tech Stack

03



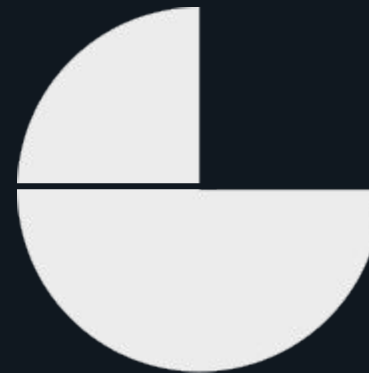
Service Flow

04



Feature Preview

05



History

1
2
3
4

Feature	Description
Multi-Stack Architecture	<ul style="list-style-type: none">- 유저(Spring Boot) / 관리자(NestJS) / AI(FastAPI) 서버를 분리하여 목적에 최적화된 아키텍처 설계 및 구축
Advanced CMS	<ul style="list-style-type: none">- RAG(Vector DB) 기반 AI 어시스턴트 구현- 사용자 행동 분석 및 통계 시스템 구축- Lexical 기반 커스텀 에디터
Performance & SEO	<ul style="list-style-type: none">- Next.js의 SSR/ISR 렌더링 최적화- SEO를 통한 사용자 유입 극대화
On-premise Infrastructure	<ul style="list-style-type: none">- Raspberry Pi와 Docker Compose 기반의 다중 서비스 인프라- GitHub Actions를 통한 CI/CD 환경 구축- Prometheus/Grafana를 이용한 리소스 수집 및 시각화

FRONT

ADMIN

- React
- React Query
- MUI (Material UI)
- Axios
- Lexical Editor

USER

- Next.js
- PWA(Manifest)
- SEO
 - Sitemap
 - RSS
 - Meta tag
- Module.css
- Styled-components

BACK

ADMIN

- TypeScript
- NestJS
- TypeORM
- JWT Authentication

AI

- Python
- FastAPI
- SQLAlchemy
- LangChain

USER

- Java
- Spring Boot
- Spring Data JPA

INFRA

DB / Cache / Messaging

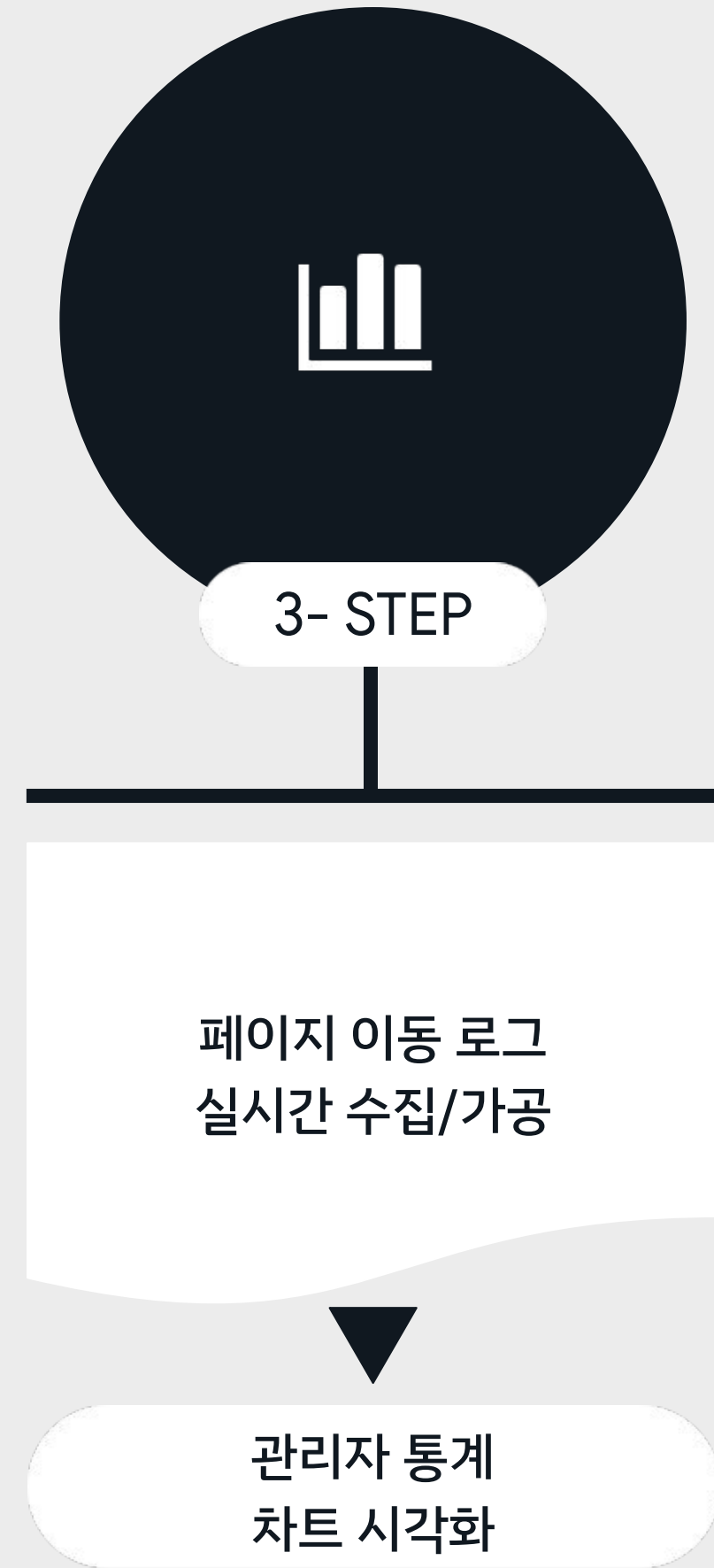
- PostgreSQL (RDBMS)
- Qdrant (Vector DB)
- Redis
- RabbitMQ

Runtime / Deployment

- Raspberry Pi
- Docker Compose
- Nginx
- AWS Route 53 (DDNS)
- GitHub Actions (CI/CD)

Monitoring

- Prometheus
- Grafana



Qdrant

Welcome
Console
Collections
Tutorial
Datasets
Access Tokens

bluecool_posts

Points Info Cluster Search Quality Snapshots Visualize Graph

Point 02c77cf5-be44-4dcd-8556-5b3748bf0748

Payload

```

{
  "page_content": "Title: 🔑 SSH & SFTP 기초 가이드 - 서버 접속과 파일 전송 SSH 접속 ... "
  "metadata": {
    "post_id": 15
    "title": "🔑 SSH & SFTP 기초 가이드 - 서버 접속과 파일 전송"
    "description": "서버 운영에 필수적인 두 가지 작업인 SSH와 SFTP의 기본 사용법을 정리했다. SSH는 ... "
    "slug": "ssh-sftp-guide"
    "category": "Tips"
    "published_at": "2025-08-09T09:06:01.070Z"
    "keywords": [
      0: "SSH"
      1: "SFTP"
      2: "FileZilla"
    ]
    "Header 2": ""
  }
}

```

Vectors:

Default vector Length: 768 [Copy](#) [Open Graph](#) [Find Similar](#)

Point 0624c7ce-cc4c-4729-b507-10fd7de67310

Payload

```

{
  "page_content": "Title: 🖨️ DNS부터 렌더링까지: 웹페이지가 열리는 과정 Section: 1. DN ... "
  "metadata": {

```

Qdrant v116.3

AI ASSISTANT

01. 콘텐츠 임베딩

블로그 글 작성 시 텍스트를 임베딩 벡터로 변환하여 검색 가능한 형태로 전처리

02. 메타데이터 설계

Qdrant를 활용한 임베딩 벡터 저장 및 메타데이터 구성을 통한 검색 필터링

03. 지식 기반 데이터

콘텐츠 벡터와 메타데이터를 통합 관리하여 RAG 기반 질의응답이 가능한 지식 DB



표민
ADMIN



+ 새 글 작성

대시보드

- 통계
- 글 관리
- 카테고리 관리
- 댓글 관리
- 계정 관리
- 방명록 관리
- 웹 사이트

오늘 방문자 (UV)

5

오늘 페이지뷰 (PV)

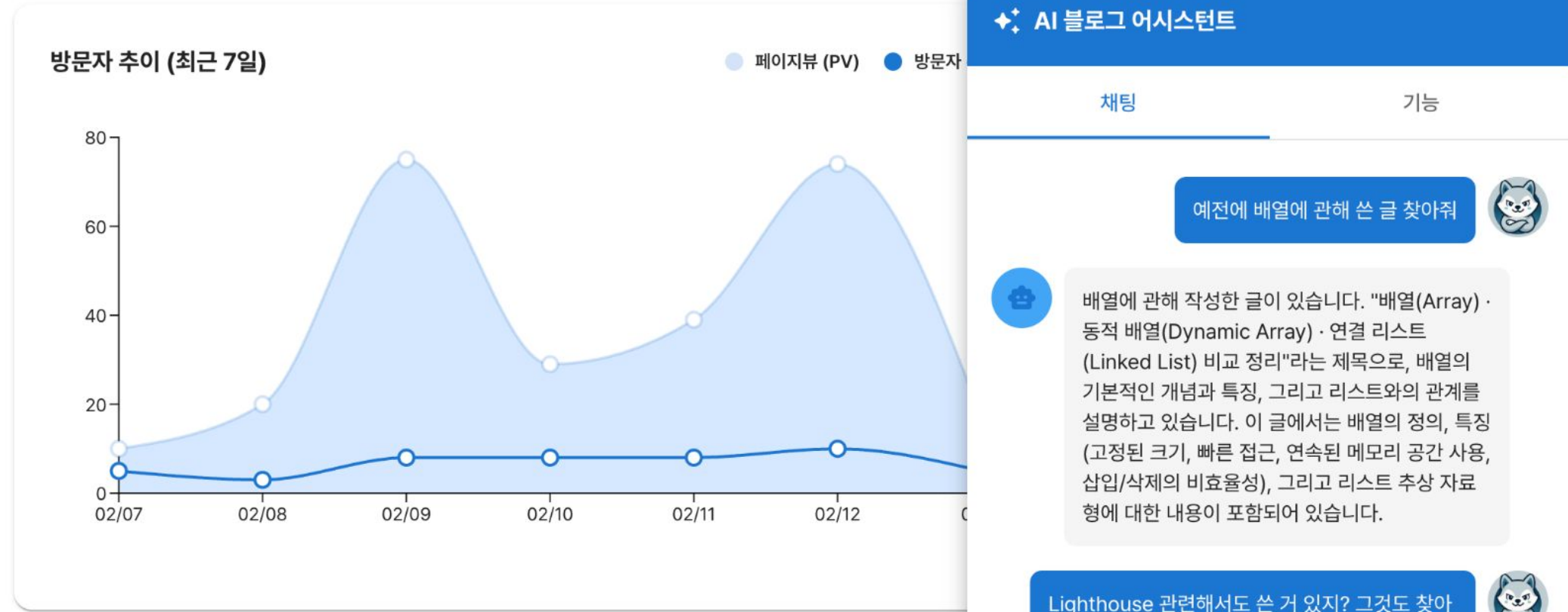
17

대기 중인 댓글

35

전체 게시글

43



- ### 🔥 인기 게시글 (Top 3)
- ☕ JS가 싱글 스레드로 동시성을 처리하는 방법 (이벤트 루프 & 블로킹)
조회수: 20회
 - 🌱 NestJS와 모듈 시스템 이해하기
조회수: 16회
 - 👁 한 눈에 정리하는 JS 비동기 3단계: Callback, Promise, Async/Await

AI 블로그 어시스턴트

채팅 | 기능

예전에 배열에 대해 쓴 글 찾아줘

배열에 대해 작성한 글이 있습니다. "배열(Array) · 동적 배열(Dynamic Array) · 연결 리스트(Linked List) 비교 정리"라는 제목으로, 배열의 기본적인 개념과 특징, 그리고 리스트와의 관계를 설명하고 있습니다. 이 글에서는 배열의 정의, 특징(고정된 크기, 빠른 접근, 연속된 메모리 공간 사용, 삽입/삭제의 비효율성), 그리고 리스트 추상 자료형에 대한 내용이 포함되어 있습니다.

Lighthouse 관련해서도 쓴 거 있지? 그것도 찾아줘

네, Lighthouse 관련해서 작성한 글이 있습니다.

M-O에게 무엇이든 물어보세요!

AI ASSISTANT

01. AI 서버 구성

FastAPI 기반 AI 서버 구축
LangChain을 활용한
Gemini API 연동

02. Event-Driven

RabbitMQ 기반 비동기
메시징 처리 및 Redis를
활용한 작업 상태 캐싱 구현

03. 프롬프트 설계

기능별 요구사항을 바탕으로
프롬프트 구조화를 통해
응답 품질 개선



표민
ADMIN



+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트

오늘 방문자 (UV)

5



오늘 페이지뷰 (PV)

17



대기 중인 댓글

35

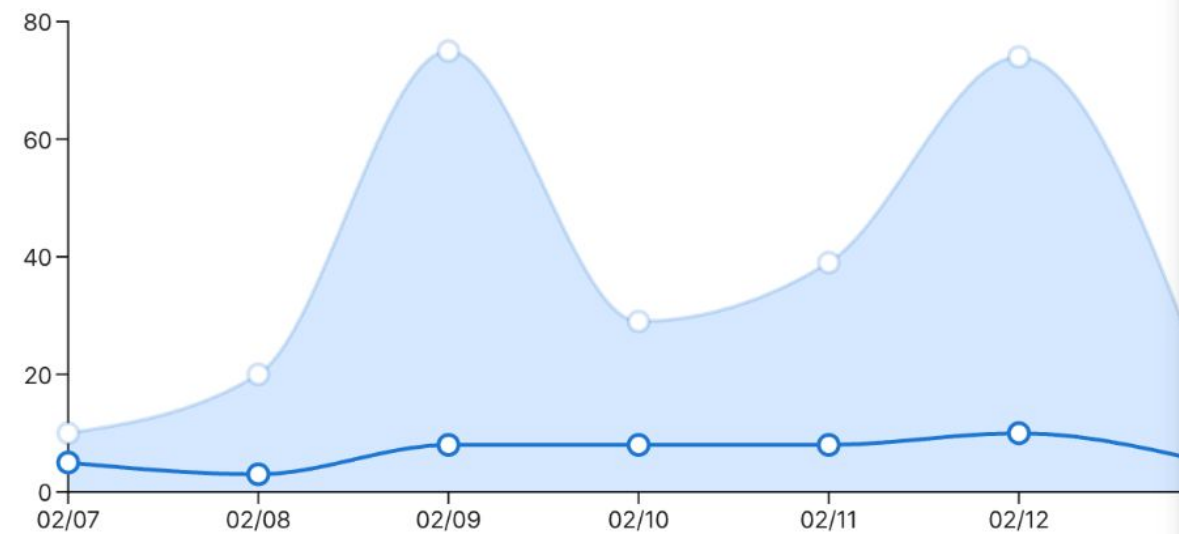


전체 게시글

43



방문자 추이 (최근 7일)



🔥 인기 게시글 (Top 3)

1. 🍷 JS가 싱글 스레드로 동시성을 처리하는 방법 (이벤트 루프 & 블로킹)

조회수: 20회

2. 🌱 NestJS와 모듈 시스템 이해하기

조회수: 16회

3. 📞 한 눈에 정리하는 JS 비동기 3단계: Callback, Promise, Async/Await

AI 블로그 어시스턴트

채팅

기능

추천 주제

[TypeScript]

⚙️ TypeScript 고급 타입: 조건부 타입과 맵드 타입 완전 정복

SLUG 생성

JS가 싱글 스레드로 동시성을 처리하는 방법

js-single-thread-concurrency

글 요약

1. 싱글 스레드의 딜레마
자바스크립트는 싱글 스레드(Single Thread) 기반의 언어이다. 쉽게 말해 한 번에 하나의 일밖에 처리를 못 한다는 뜻이다.

AI ASSISTANT

01. 글 주제 추천

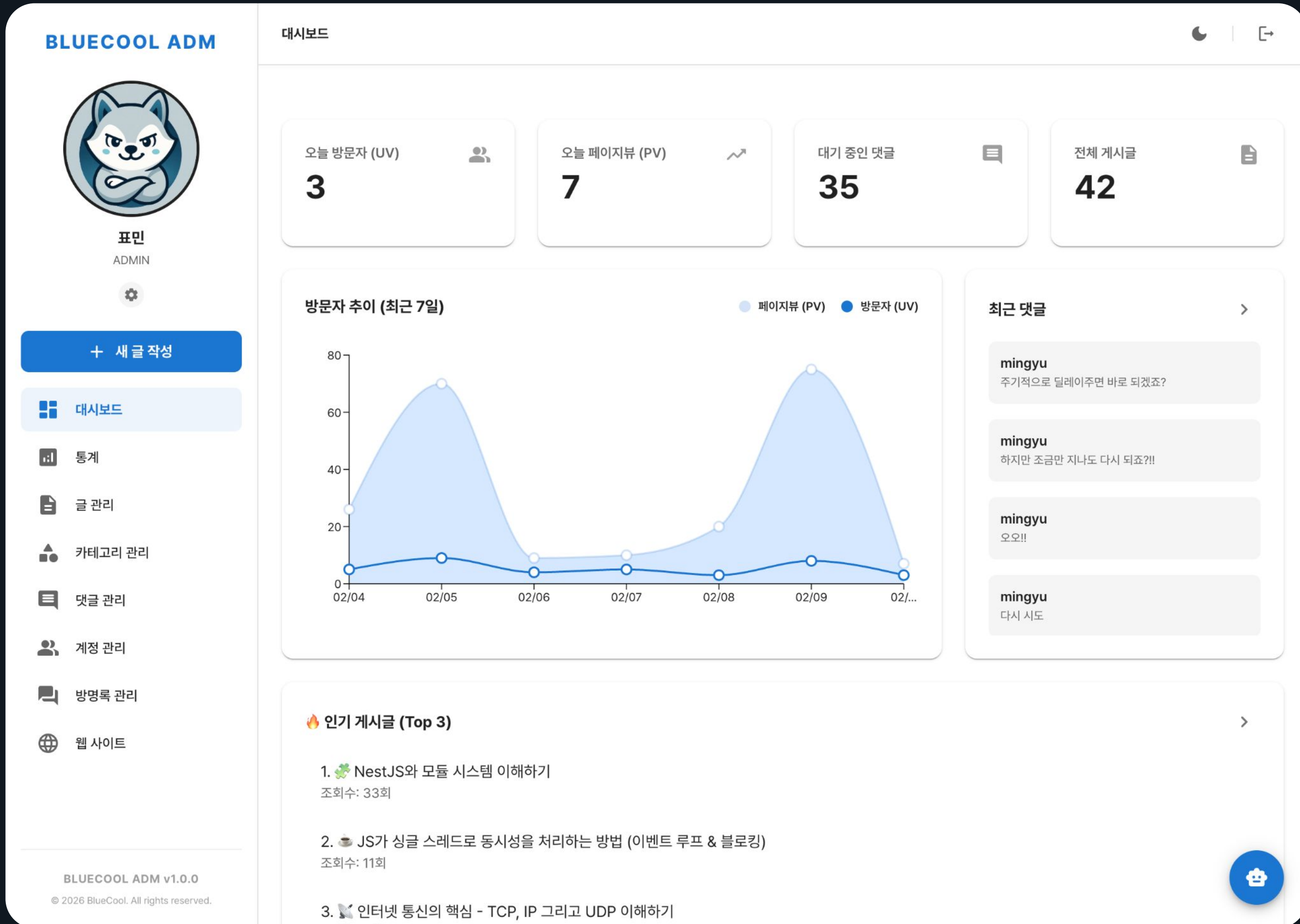
블로그 운영 이력을 기반으로 다음에 작성할 콘텐츠 주제를 AI가 제안

02. 영문 Slug 변환

제목을 입력 시 URL 최적화를 고려한 영문 slug로 자동 변환

03. SEO 요약 생성

글 내용을 분석해 SEO 및 메인 페이지 노출을 위한 200자 내외 요약 제공



DASHBOARD

01. 데이터 시각화

MUI X-Charts를 활용한 방문자 추이 및 콘텐츠 통계

02. 상태 관리

React Query를 통해 실시간 데이터 페칭 및 최적화된 캐싱 처리

03. 운영 지표

서비스 전반의 핵심 데이터를 한번에 파악할 수 있도록 UI 구축

표민
ADMIN

+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트

블로그의 흐름과 독자의 반응을 분석해보세요.

방문자 추이와 인기 콘텐츠 데이터를 통해 블로그의 성장 가능성을 확인합니다.

오늘 방문자 (UV)



3명

-62.5% 전일 대비

오늘 조회수 (PV)



7회

-90.7% 전일 대비

평균 체류 시간



19초

-69.9% 전일 대비

이탈률

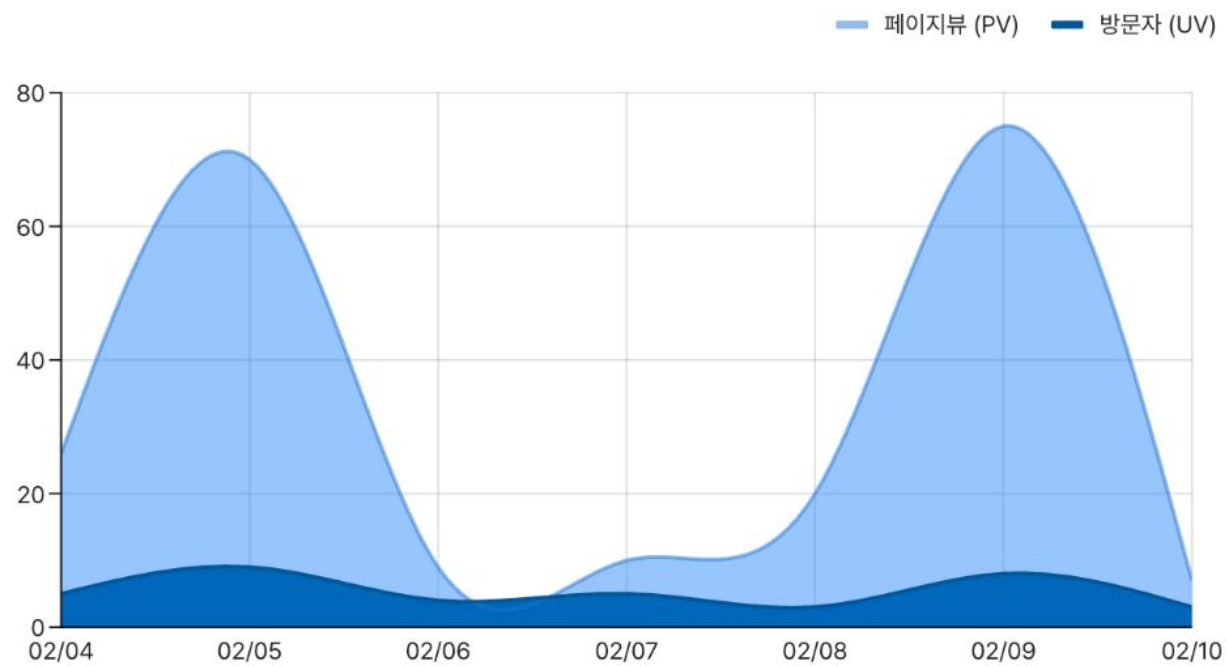


66.7%

+183.8% 전일 대비

방문자 추이

최근 7일 | 최근 30일



실시간 인기 콘텐츠

- 1 NestJS와 모듈 시스템 이해하기
33 views today
- 2 JS가 싱글 스레드로 동시성을 처리...
11 views today
- 3 인터넷 통신의 핵심 - TCP, IP 그리...
2 views today
- 4 배열(Array) · 동적 배열(Dynamic ...
2 views today
- 5 운영 환경까지 고려한 JPA 설정 가...
2 views today

콘텐츠별 상세 성과 분석

글 제목

조회수(PV)

방문자(UV) | 게시일



ANALYTICS

01. 핵심 지표 모니터링

방문자, 조회수, 평균 체류시간 등 주요 지표 실시간 집계

02. 성장 추세 계산

전일 대비 증감률 및 차이값을 계산하여 성장 추세 확인

03. 랭킹 시스템

실시간 조회수 기반의 Top 5 인기 게시글 랭킹 제공



표민
ADMIN



+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트

콘텐츠별 상세 성과 분석

글 제목	조회수(PV)	방문자(UV)	게시일
NestJS와 모듈 시스템 이해하기 nestjs-module-system	37	9	2026. 2. 1. 오후 11:24:54
한 눈에 정리하는 JS 비동기 3단계: Callback, Promise, Async/Await javascript-async-callback-promise-async-await	16	7	2026. 1. 27. 오후 9:13:17
JS가 싱글 스레드로 동시성을 처리하는 방법 (이벤트 루프 & 블로킹) js-single-thread-concurrency-event-loop-blocking	11	4	2026. 2. 5. 오후 9:33:11
비선형 자료구조의 핵심 - 트리(Tree) 구조 완전 정복 mastering-tree-data-structure	3	3	2026. 1. 24. 오후 5:50:37
배열(Array) · 동적 배열(Dynamic Array) · 연결 리스트(Linked List) array-vs-dynamic-array-vs-linked-list	3	3	2025. 6. 25. 오후 4:58:04

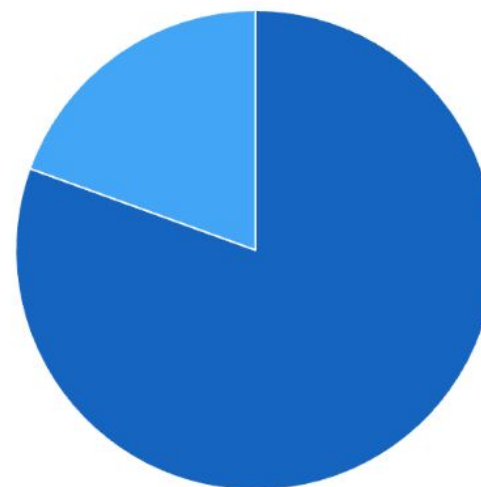
페이지 당 행: 5 1-5 of 42

유입 경로 (Referrer)



Others Direct Google OKKY

접속 기기 (Device)



Desktop Mobile

ANALYTICS

01. 콘텐츠 분석

효율적인 처리를 위한
Pagination 구현

02. 다차원 데이터

유입 경로와 접속 기기
데이터를 그룹화 및
정제하여 시각화



표민
ADMIN



+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트

차곡차곡 쌓인 지식의 기록들을 살펴보세요.

지금까지 작성한 소중한 생각들을 한눈에 확인하고 관리할 수 있습니다.

+ 새 글 작성

전체

공개됨

초안

보관

전체 ▾

제목 혹은 내용 검색



YYYY-MM-DD



| YYYY-MM-DD



검색 결과 총 42개의 게시물이 발견되었습니다.



번호	제목	상태	작성일	관리
68	☕ JS가 싱글 스레드로 동시성을 처리하는 방법 (이벤트 루프 & 블로킹)	공개	2026-02-05	
66	🌱 NestJS와 모듈 시스템 이해하기	공개	2026-02-01	
65	☎ 한 눈에 정리하는 JS 비동기 3단계: Callback, Promise, Async/Await	공개	2026-01-27	
64	🌿 비선형 자료구조의 핵심 - 트리(Tree) 구조 완전 정복	공개	2026-01-24	
63	📘 TypeScript를 사용하는 이유	공개	2026-01-21	
62	🐛 NestJS 증분 빌드 메커니즘 이해와 캐시 이슈 정리	공개	2026-01-19	
58	🦊 자바스크립트와 Long 타입 ID (feat. NestJS)	공개	2026-01-14	
53	👤 Lombok의 생성자 어노테이션	공개	2026-01-10	

POST

01. 다중 필터링

글 상태, 카테고리, 날짜별
조건 검색으로 관리 편의성
극대화

02. 검색

제목 및 본문 기반의 실시간
검색 기능을 통한 빠른
콘텐츠 접근

03. 콘텐츠 제어

리스트 내에서 직접 수정 및
삭제가 가능한 원스톱 관리
기능 제공



표민
ADMIN



+ 새 글 작성

🏠 대시보드

📊 통계

📄 글 관리

📁 카테고리 관리

💬 댓글 관리

👤 계정 관리

🗨️ 방명록 관리

🌐 웹사이트

📌 Spring으로 알아보는 의존성 주입(DI)

↩️ ↪️ A 본문 ▾ B I U 주 A 🎨 <> 🔗 - 🖼️ ☰ ☷ ☸ Javascript ▾

final 필드를 사용할 수 있어 불변성을 지킬 수 있으며 테스트 코드에서 생성자만 호출하면 되기 때문에 테스트도 용이하
다. (Lombok 사용 시 [@RequiredArgsConstructor](#)로도 가능)

*필드 주입

```
@Service
public class PostService {

    @Autowired
    private PostRepository postRepository;
}
```

코드가 굉장히 간단해지지만 필드가 final이 될 수 없고 테스트 시 Mock 주입이 불편하여 요즘은 **지양**하는 방식으로 취급된다. (DI 컨테이너 없이 순수 자바 코드로 생성하기도 어렵다)

*Setter 주입

```
@Service
public class PostService {

    private PostRepository postRepository;
```

👁️ 미리 보기

📌 임시 저장

✅ 작성 완료



POST

01. Lexical Editor

Meta의 에디터 엔진을
분석하여 블로그 전용
툴바 및 플러그인 직접 구현

02. 임시 저장 시스템

작성 중인 데이터 손실을
방지하기 위한 안정적인
임시 저장 로직 구축

03. 상세 설정 모달

글 공개 여부, 카테고리
지정 등 게시글 속성을
제어하는 직관적인 설정 UI

실시간 미리보기
미리보기 닫기

ABOUT POSTS GUESTBOOK
🎵 🌙

2025-09-21
Network

📄 DNS부터 렌더링까지: 웹페이지가 열리는 과정

브라우저에 www.pyomin.com을 입력하면...

브라우저 주소창에 www.pyomin.com을 입력하고 엔터를 치면 단순히 페이지가 열리는 것 같지만 그 뒤에서는 복잡한 과정이 순식간에 일어난다. 그 과정을 한 단계씩 따라가 보자.

1. DNS 조회

컴퓨터는 www.pyomin.com 같은 도메인을 직접 이해하지 못한다. IP 주소로 통신하기 때문인데 따라서 먼저 **DNS(Domain Name System)** 조회가 일어난다.

브라우저는 OS에 www.pyomin.com의 IP를 요청하고 OS는 먼저 **로컬 캐시**를 확인한다. 캐시에 존재하지 않는 경우 ISP(인터넷 서비스 제공 업체)의 DNS 서버로 요청을 보내고 필요한 경우 루트 DNS 서버 → TLD 서버 → 권한이 있는 네임서버까지 찾아가며 최종 IP 주소를 알아낸다.

2. TCP/IP 연결

POST

01. 실시간 미리보기

별도의 창 이동 없이
iframe을 이용하여 실제
블로그 화면을 실시간 확인

02. 검증 최적화

게시 전 검색 엔진 최적화 및
레이아웃을 실제 유저 환경과
동일하게 미리 검수



표민
ADMIN



+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트

블로그의 목차를 정리해보세요.

카테고리를 추가하거나 수정하여 글을 찾기 쉽게 분류할 수 있습니다.

카테고리 수정

삭제

상위 카테고리

Backend

카테고리 이름

NestJS

URL 슬러그 (Slug)

nestjs

영문 소문자와 하이픈(-)만 사용

저장

초기화 (새로 만들기)

카테고리 목록

Language
/language

Backend
/backend

Spring
/spring

NestJS
/nestjs

JPA
/jpa

Frontend
/frontend

Dev
/dev

DevOps
/devops

CS
/cs

CATEGORY

01. 2-Pane 레이아웃

좌측(생성/수정)과 우측(목록 조회)으로 기능을 분리하여 관리 몰입도 향상

02. 계층 구조 시각화

상위-하위 관계가 명확히 구분된 리스트 인터페이스로 서비스 구조 시각화

표민
ADMIN

+ 새 글 작성

대시보드

통계

글 관리

카테고리 관리

댓글 관리

계정 관리

방명록 관리

웹 사이트



전체 게시글에 등록된 댓글을 확인하고 관리합니다.

방문자들의 소중한 피드백을 확인하고 답글을 남겨보세요.

전체 댓글

답변 대기

답변 완료

익 익명의고릴라 삭제됨

2025-11-01

사용자 또는 관리자에 의해 삭제된 댓글입니다.

원문: [🔗 CI/CD 오류: GHCR을 이용한 GitHub Acti...](#)

저 저 사모예드인가요?

2025-10-04

ㅋㅋ

원문: [🔗 운영 환경까지 고려한 JPA 설정 가이드](#)

익 익명의사모예드 숨김

2025-10-04

잘 정리된 글 감사합니다

원문: [🔗 운영 환경까지 고려한 JPA 설정 가이드](#)

익 익명의개발자 삭제됨

2025-09-27

사용자 또는 관리자에 의해 삭제된 댓글입니다.

원문: [🔗 인터넷 통신의 핵심 - TCP, IP 그리고 UDP ...](#)

COMMENT

01. 워크플로우 최적화

답변 대기 및 완료 필터링을 통해 미응답 댓글을 즉각 파악하고 관리

02. 다이렉트 인터랙션

관리자 페이지 내에서 바로 답글 작성이 가능한 기능으로 사용자 소통 강화

03. 댓글 제어

개별 댓글의 상태 변경 및 삭제를 테이블 리스트에서 즉각 처리

표민
ADMIN

+ 새 글 작성







- 대시보드
- 통계
- 글 관리
- 카테고리 관리
- 댓글 관리
- 계정 관리**
- 방명록 관리
- 웹 사이트

운영 계정 및 보안 설정

등록된 모든 관리자 계정을 확인하고 권한을 수정하거나 새로운 멤버를 추가하세요.

+ 계정 추가

🔍 아이디, 이름, 닉네임 검색

계정 정보	활동명	권한	최근 접속일	설정
박병찬 byoengchan	두존쿠헌터	USER	기록 없음	
표민 pmini12	BLUECOOL	ADMIN	2026. 2. 10. 오후 1:35:04	
이민규 mingyu	영원한스승님	USER	2026. 2. 9. 오후 11:39:46	
김태원  taewon	AI조련사	USER	2026. 2. 7. 오후 5:01:01	
이창하 changha	똑똑박사	USER	2026. 1. 29. 오후 9:32:05	

페이지당 행 수: 10 ▼ 1-5 / 전체 5 < >

ADMIN

01. 권한 검증

JWT 기반으로 사용자 권한을 실시간 검증하고 메뉴 접근 제어

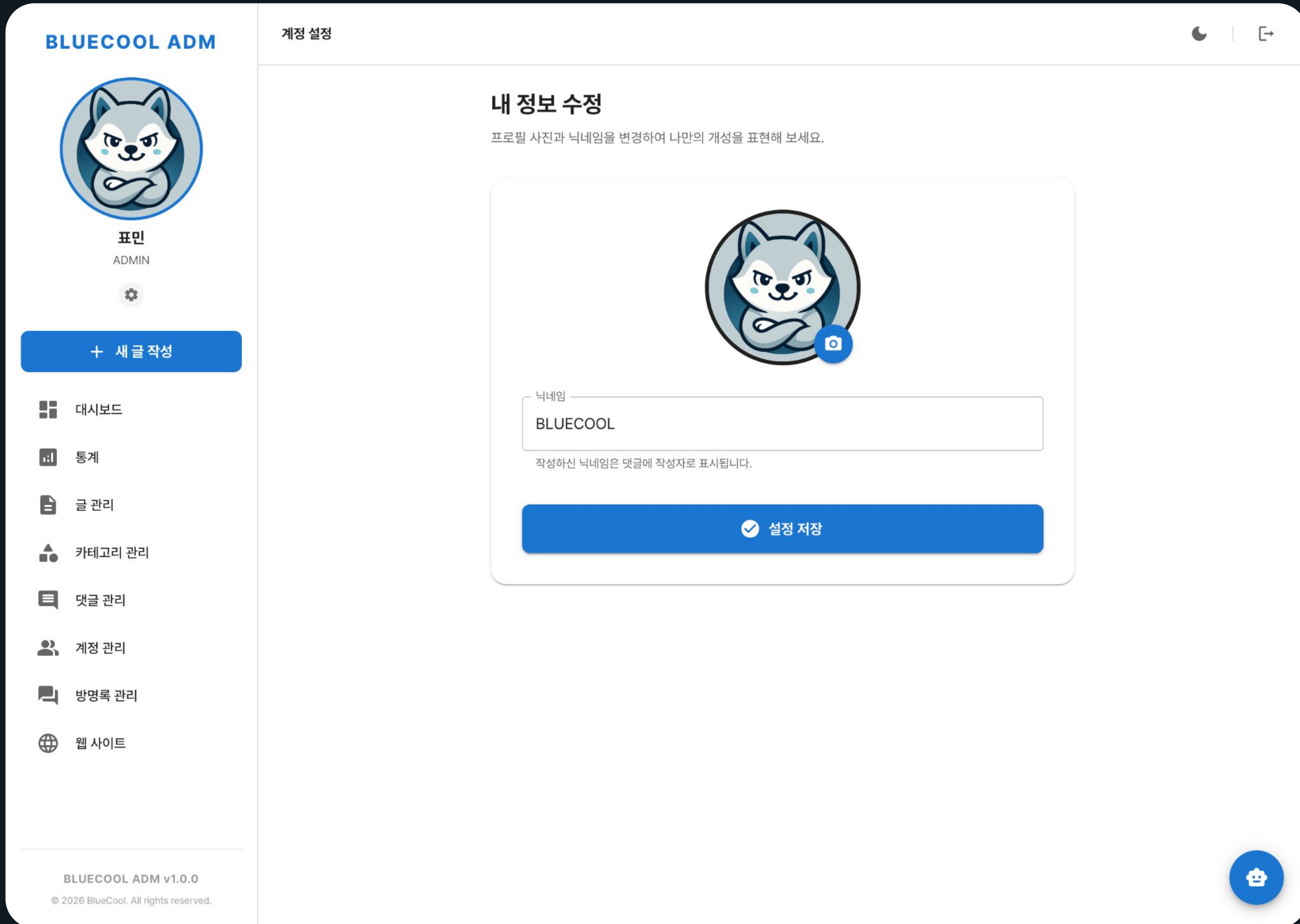
02. 계정 보안 모니터링

계정별 권한, 최근 접속일, 계정 잠금 상태를 가시화하여 보안성 강화

03. 효율적인 관리 UI

통합 검색 및 모달 기반의 신규 계정 생성 워크플로우 구축





PROFILE

01. 프로필 변경

관리자 활동명과 프로필 사진을 자유롭게 변경하여 서비스 정체성 관리

02. 실시간 이미지 갱신

이미지 저장 시 클라이언트 사이드 캐시 무효화를 통해 프로필 영역 즉시 업데이트





ABOUT

POSTS

GUESTBOOK



Hello, World!

블로그에 방문하신 여러분
환영합니다 :D



Daily Pick 👁️

매일 새롭게 만나는 세 가지 주제



React



자료구조



Java

최신 글 ✨

새로 올라온 글들을 확인해보세요



MAIN

01. 사용자 행동 분석

특정 사용자의 유입부터 이탈까지의 흐름을 수집하여 운영 인사이트 도출

02. Advanced UX

다크모드 구현 및 앱과 유사한 사용자 경험 (Manifest) 구축

03. 콘텐츠 레이아웃

매일 새로운 카테고리 선정 및 최신 글, 트러블 슈팅 전용 섹션 배치



ABOUT

POSTS

GUESTBOOK



방명록 남기러 가기

4 Comments - powered by utteranc.es



BlueCool12 commented on 2025년 6월 2일

Owner + 😊

ㄸ^..ㅅ.^ㄸ 반갑습니다 ㄸ^..ㅅ.^ㄸ



1



1



2



1



2



2



1



1



TaeWonKim2028 commented on 2025년 6월 10일

+ 😊

블로그 잘 보고 갑니다 ^^*



1



TaeWonKim2028 commented on 2025년 6월 17일

+ 😊

=^w^=



1



nullisdefined commented 3 months ago

+ 😊

주요 글자번호 갱신

GUESTBOOK

01. Utterances 통합

GitHub Issue 기반의 댓글 시스템을 활용하여 별도의 DB 관리 없이 방명록 구현

02. 개발자 맞춤형 UI

마크다운 형식을 지원하여 피드백에 최적화된 소통 공간 제공



🍷 JS가 싱글 스레드로 동시성을 처리하는 방법 (이벤트 루프 & 블로킹)

JavaScript

1. 싱글 스레드의 딜레마 자바스크립트는 싱글 스레드(Single Thread) 기반의 언어이다. 쉽게 말해 한 번에 하나의 일밖에 처리할 수 없다는 뜻이다. 그런데 우리는 브라우저에서 유튜브를 보면서 댓글도 달고 Node.js 서버는 수천 명의 요청을 동시에 처리하는 것처럼 보인다. 어떻게 가능한 것일까? 그 비결은 블로킹/논 블로킹(Blocking/Non-blocking) I/O와 이를 지휘하는 이...

2026.02.05

🌱 NestJS와 모듈 시스템 이해하기

NestJS

1. 모듈의 정의 소프트웨어 공학에서 모듈(Module)은 거대한 프로그램을 구성하는 독립적인 단위이자 기능적 부품을 의미한다. 레고로 집을 만든다고 상상해 보자. 창문, 벽, 문 블록을 따로 만든 뒤 조립한다. 이때 중요한 것은 각 블록이 서로 맞물리는 표준 규격(interface)을 가졌다는 점이다. 모듈화를 하는 이유는 크게 3가지다. 유지 보수의 효율성 (Manageability) 특정 모듈이...

2026.02.01

📞 한 눈에 정리하는 JS 비동기 3단계: Callback, Promise, Async/Await

JavaScript

자바스크립트는 싱글 스레드로 동작하는 언어이다. 따라서 한 번에 하나의 작업만 처리할 수 있다. 하지만 우리가 웹서핑을 할 때 이미지를 불러오면서 동시에 클릭도 할 수 있는 이유는 바로 비동기 처리 덕분이다. 자바스크립트에서 이 비동기 처리가 어떻게 발전해 왔는지 흐름을 알아보자. 1. 비동기의 시작: 콜백 함수(Callback) 콜백 함수는 특정 작업이 끝난 뒤에 실행하기 위해 나...

2026.01.27

🌳 비선형 자료구조의 핵심 - 트리(Tree) 구조 완전 정복

자료구조

배열, 스택, 큐 같은 자료구조가 데이터를 일렬로 세운 선형 구조였다면 오늘 다룰 트리(Tree)는 조금 다르다. 데이터를 계층적으로 쌓아 올리는 비선형 구조라 처음 접하면 다소 낯설게 느껴질 수 있지만 사실 트리는 우리가 매일 사용하는 파일 시스템, 웹 페이지의 HTML DOM, 데이터베이스 인덱스 등에 이미 깊숙이 자리 잡고 있다. 트리(Tree)란? 트리는 *노드(Node)들이 부...

2026.01.24

📖 TypeScript를 사용하는 이유

TypeScript

0. TypeScript를 사용하는 이유 TypeScript를 공부하기 전에 먼저 한 가지 질문을 던져볼 필요가 있다. JavaScript로도 잘 동작하는데 왜 굳이 TypeScript를 사용해야 할까? 이 글에서는 본격적으로 TypeScript 문법을 다루기 전에 TypeScript가 무엇인지, 그리고 어떤 문제를 해결하기 위해 등장했는지를 정리해보려고 한다. TypeScript를 단순한 문법이 아닌, 무엇을 위한 ...

2026.01.21

🐛 NestJS 증분 빌드 메커니즘 이해와 캐시 이슈 정리

트러블슈팅

POSTS

ALL

LANGUAGE

BACKEND

FRONTEND

DEV

DEVOPS

CS

01. 무한 스크롤

TanStack Query의 useInfiniteQuery를 활용하여 구현

02. 데이터 페칭 최적화

사용자의 스크롤 위치에 맞춰 다음 데이터를 동적으로 호출

03. 카테고리 네비게이션

직관적인 탭 UI를 통해 관심 있는 주제의 글만 빠르게 필터링하여 조회



```
// main.ts
if (!(BigInt.prototype as any).toJSON) {
  Object.defineProperty(BigInt.prototype, 'toJSON', {
    value() {
      return String(this);
    },
    configurable: true,
    enumerable: false,
    writable: true,
  });
}
```

TYPESCRIPT

이렇게 하면 전역적으로 모든 BigInt가 JSON 직렬화 시 문자열로 변환되어 정밀도 유실 문제를 원천 봉쇄할 수 있다.

단, 서버에서 보낼 때는 문자열로 자동 변환되지만 프론트엔드에서 문자열로 보낸 ID를 서버에서 숫자로 계산해야 한다면 BigInt()를 통해 숫자로 변환하는 과정이 필요하다.

이전 글



👤 Lombok의 생성자 어노테이션

다음 글



🐛 NestJS 증분 빌드 메커니즘 이해와 캐시 이슈 ...



첫 댓글을 남겨보세요! (= 'X' =)



DETAIL

01. Google Adsense

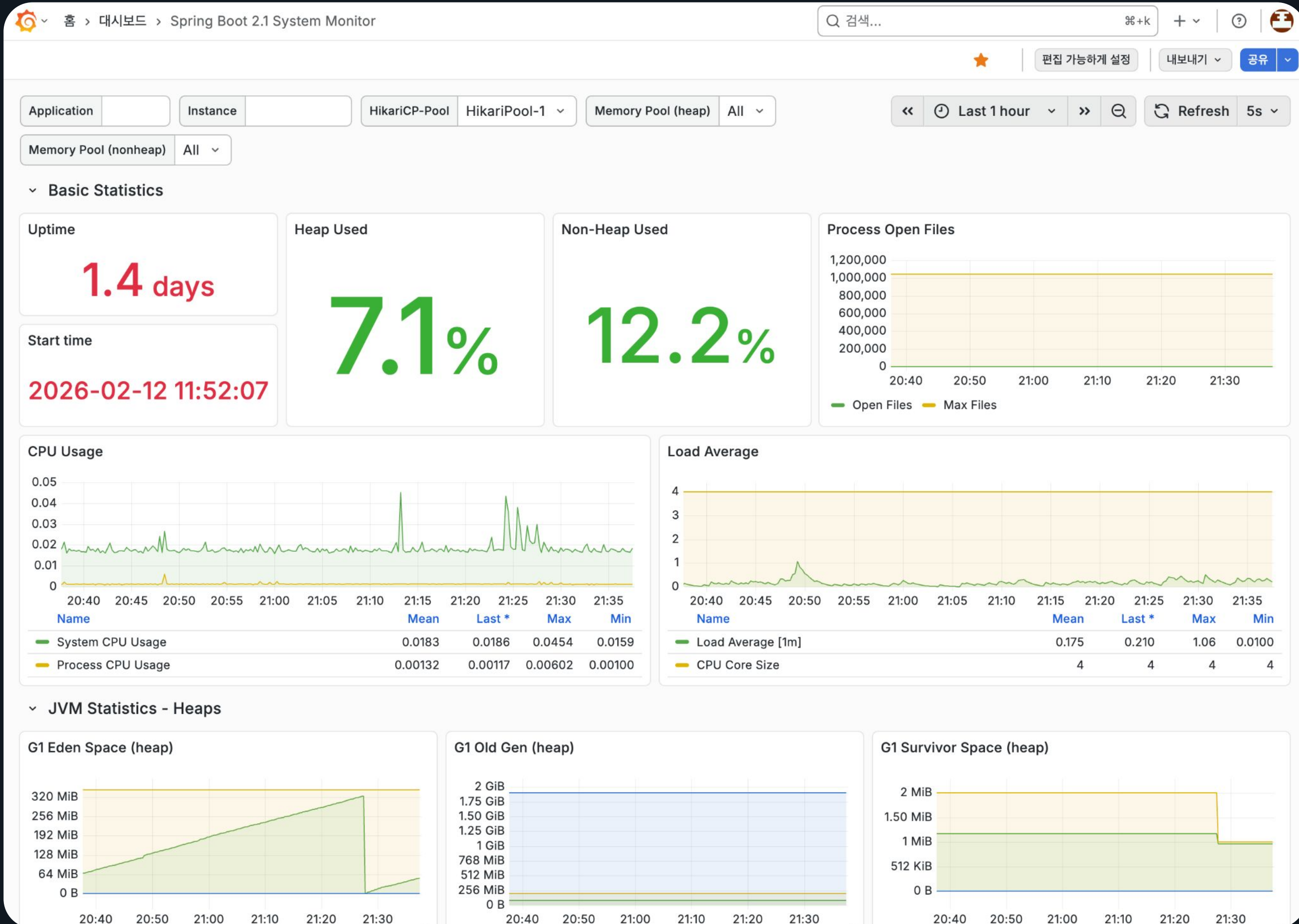
서비스 운영 지속성을 위한
구글 애드센스 광고 최적화
배치 및 수익화 기반 마련

02. Shiki 코드 스타일

실제 IDE와 유사한 코드
하이라이팅을 적용하여
가독성 극대화

03. 플로팅 인터랙션

화면 하단에 고정된 댓글
버튼을 배치하여 즉각적으로
소통할 수 있는 편의성 제공



MONITORING

01. 메트릭 수집 환경

Prometheus를 활용한
서버 및 컨테이너 메트릭
수집 환경 구성

02. 대시보드 시각화

Grafana 기반의 서비스 및
인프라 상태를 한눈에
확인할 수 있는 대시보드

03. 운영 안정성 및 성능

수집된 메트릭을 기반으로
리소스 병목 구간을
분석하고 지속적으로 점검

HISTORY

VERSION	AUTHOR	DESCRIPTION	DATE
0.1	MIN PYO	라즈베리 파이 서버 환경 설정 Docker/Linux 최적화	06.12.
0.2	MIN PYO	AWS Route 53 연동 DDNS 구축을 통한 외부 도메인 연결	06.28.
0.3	MIN PYO	Spring Boot 기반 유저 API 서버 사용자 행동 로깅 DB 구축	07.15.
0.4	MIN PYO	Next.js 기반 메인 페이지 방명록 구현	08.20.
0.5	MIN PYO	무한 스크롤 글 목록, 다크모드 PWA 및 SEO 최적화	09.12.
0.6	MIN PYO	NestJS 기반 어드민 API 서버 JWT 권한 관리 시스템 구축	10.25.
0.7	MIN PYO	대시보드 시각화 계정/댓글/카테고리 관리 페이지 구현	11.18.
0.8	MIN PYO	Lexical 기반 커스텀 에디터 개발 실시간 미리보기 기능 구현	12.25.
0.9	MIN PYO	GitHub Actions CI/CD 구축 인프라 - 서버 - UI 최종 안정화	01.17.

THANK YOU

[Link: https://pyomin.com](https://pyomin.com)

