

PNU SW학습공동체 최종보고서

Github로 파일업로드하여 제출

1. 프로젝트 소개

가. 배경 및 필요성

개발자들은 코딩테스트 준비에 큰 부담을 느끼며 비효율적인 방식으로 스터디를 꾸준히 진행하지 못한다는 현실을 목격하고 체감하였습니다. 특히 40명 이상의 개발자 및 대학생 인터뷰를 통하여 아래와 같은 두 가지 문제점을 도출하였습니다.

첫째, 혼자서 꾸준히 학습하기 어렵다는 점입니다. 알고리즘 학습은 학과 공부와는 다른 성격을 가지며, 개인이 체계 없이 문제를 풀다보면 올바른 공부 방법도 모를 뿐더러 금세 흥미를 잃는 경우가 많았습니다.

둘째, 기존 스터디 방식의 한계입니다. 많은 스터디가 단순 인증 중심으로 운영되고 있으며, 참여자들이 어떤 방식으로 문제를 해결했는지에 대한 구체적인 정보가 공유되지 않는 경우가 많았습니다. 또한 풀이 공유나 회고 문화가 활성화되어 있는 스터디는 드물고, 공유 환경이 충분히 갖추어지지 않아 학습의 깊이를 더하는 데 어려움이 있었습니다.

이러한 문제를 해결하기 위하여 '코드몬스터' 프로젝트를 기획하게 되었으며, 최소 기능부터 빠르게 구현한 후 사용자들을 모아 실제 사용자 기반의 데이터를 바탕으로 지속적인 개선을 이어나가고 있습니다.

나. 개발목표 및 주요내용, 세부내용 등

‘코드몬스터’는 사용자가 주어진 날짜에 코딩테스트 문제를 풀고, 그 풀이를 기록하고 공유함으로써 지속적인 학습을 이어갈 수 있도록 돕는 커뮤니티 기반 플랫폼입니다. 사용자가 처음 가입하면 실력 수준에 따라 적절한 스터디 팀에 배정되며, 해당 팀의 진행 일정에 따라 맞춤형 문제를 추천받고 풀이를 시작할 수 있도록 설계되어 있습니다. 이를 통해 개인의 성향과 수준에 맞춘 학습을 유도하며, 꾸준한 참여를 도와주는 환경을 제공합니다. 현재 약 150여 명의 대학생, 취준생 및 현직 개발자가 플랫폼에 참여하고 있으며, 문제 풀이 기록, 참여 빈도, API 사용량 등 다양한 사용자 데이터를 수집할 수 있는 구조를 갖추고 있습니다. 이 프로젝트의 핵심 개발 목표는 이러한 사용자 데이터를 기반으로 서비스를 개선하고, 사용자 중심의 기능 고도화를 통해 실질적인 학습 효과를 극대화하는 것입니다.

단순히 새로운 기능을 추가하는 수준에 그치지 않고, 실제 사용자의 행동 데이터를 분석하여 서비스 개선에 반영하는 것을 핵심 전략으로 삼고 있습니다. 특히 페이지 이동 기록, 사용자 행동 로그 등 다양한 데이터를 수집 및 전처리한 후, 서비스 사용량이 집중되는 구간에서의 성능 저하 원인을 파악하고 이를 해결하기 위한 구조적 개선을 진행할 예정입니다.

다. 사회적가치 도입 및 추후 계획 등

저희 서비스는 상업적 목적 없이 누구나 무료로 사용할 수 있도록 운영되고 있습니다. 저희 팀은 지난 겨울방학부터 서비스를 기획하고 개발해왔으며, 현재까지도 서버비와 유지보수 비용을 자체 부담하며 플랫폼을 운영하고 있습니다. 이러한 방식은 금전적인 부담 없이도 양질의 학습 환경을 제공할 수 있도록 합니다.

앞으로도 저희는 지속 가능한 학습 커뮤니티 구성에 가치를 두고, 누구나 성장할 수 있는 환경을 만드는 데 집중할 것입니다. 장기적으로는 지역 개발자 커뮤니티, 대학 학습 동아리 등과의 연계를 통해, 더 많은 이들이 함께 성장할 수 있는 플랫폼으로 발전시키고자 합니다.

또한 이번 여름방학 시작을 기점으로 코드몬스터 참여자들을 대상으로 설문조사를 진행하였고, 사용자 간의 깊은 교류를 위해 ‘여름-인사방’ 채널을 개설하였습니다. 이

곳에서 참여자들은 각자의 학습 목표를 재설정하고 학습 과정에서 겪는 어려움과 고민을 솔직하게 공유하며 서로의 의지를 북돋았습니다!



김민주 / 6days 🇯🇵 JAVA 25. 7. 1. 오후 11:12

안녕하세요. 저는 부트캠프를 2월에 수료하며 코몬과도 수료와 동시에 백엔드 개발자로 취준 기간을 함께 해왔습니다.

대략 3월쯤부터 2day, 6day를 병행했을 때까지는 좋았는데, 4월쯤 코몬 외 다른 코테 스테디로 매일 주5회 코테스테디를 하다 보니 코딩테스트에 회의감을 많이 느꼈습니다.

또한, 부트캠프에선 Python으로 코딩테스트를 준비해도 괜찮다 해서 부트캠프 교육 기간 내에도 Python으로 코딩테스트 주언어를 선택했습니다. 다만, Java 백엔드 개발자로 취업을 하려니 중소기업 중견기업을 목표로 하는 저에게도 언어 제한이 큰 회사 제외하고는 대부분 Java로 제한되어 있었습니다. 이러한 과정을 겪으면서 제가 코딩테스트를 준비해도 되는 걸까, 이대로 취준 기간을 보내도 되는 걸까? 라는 생각이 조금 들었습니다.

사실 목표로 하는 회사가 스펙상 정말 밑바닥에 눈도 많이 낮췄다고 생각하는데 취업시장이 많이 얼어붙은 것의 영향도 있겠지만 경쟁자들의 능력치도 상향 평준화가 되어서 신입으로 들어가기엔 어려운 시기였던 거 같습니다.

실제로 어디서 본 기사로는 IT 직군으로 신규 고용보험 증가율이 4프로대...?라는 믿기지 않는 기사를 본 순간 체감이 뼈저리게 많이 느꼈던 거 같습니다.

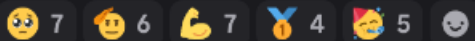
그래서 4월부터 보던 면접부터는 매번 면접 질문을 머릿속에 외워서 리마인드 하는 시간을 가졌었습니다. CS 질문들과 프로젝트 질문들을 받았던 거를 꼭 써본 결과, CS 질문 외 프로젝트에서 공통적으로 지적되는 문제점을 깨닫게 되었습니다.

따라서 사이드 프로젝트와 함께 기존 프로젝트를 한달간 리팩토링 하는 과정을 거쳤습니다. 이러면서 약 2개월간 백엔드 개발을 제대로 안 하다보니 다시금 배워가는 시간도 가지게 되었고, 현재 프로젝트에서 지적받던 부족한 역량도 채워나갔던 거 같습니다.

운이 좋게도 지난 6월 초에 면접을 봤던 회사에 최종합격을 하게 되어서 더는 지체하기엔 공백기가 두려워져 입사하게 되었습니다. 슬프게도 회사에 대해 바라는 처우협약들이 눈이 낮춰지면서 바라는 것들이 사라지더라고요. 그래서 지금 회사에 나름 만족하고 오늘 첫 출근을 하게 되었습니다!

첫 출근을 하면서도 꾸준히 자기계발을 하기 위해서 앞으로 다시 코몬으로 Java 코딩테스트를 다시 처음 class2~3으로 풀어보려고 합니다. 라이브러리가 익숙치 않아서 Python으로는 몇 줄 적고 끝나는 코드들이 Java로는 익숙치 않아서 앞으로 공부해나가보려고 합니다!

다들 어려운 시기에 힘내시고 포기하지 말고 목표로 하는 회사를 가시길 바라겠습니다. 무더운 더위 속에 다들 파이팅입니다!



이처럼 코드몬스터는 단순히 코딩 문제를 푸는 공간을 넘어, 개발자들과 취준생분들이 서로의 어려움에 공감하고, 동기를 부여하며 함께 성장할 수 있도록 돕는 활발한 커뮤니티의 장이 되고 있습니다. 저희는 연대와 공유의 문화를 통해서 참여자들이 더욱 꾸준하고 의미 있는 학습 경험을 이어나갈 수 있도록 도울 예정입니다.



이진원 / 6days 25. 7. 2. 오전 12:08

다들 요즘 잘 지내시나요?

공감되는 글이 많아서 글을 쓰지 않을 수가 없네요.

저는 문돌이고 프로그래밍을 접한지 6개월 정도 지났을 무렵인 작년 여름방학에 개발자가 되겠다는 결심을 했습니다. 근데 결심을 한 그 순간부터 무언가에 쫓기는 듯한 느낌이 들었습니다. 실제로 동나이대 전공자들에 비해서 모르는게 너무 많았고 빨리 따라잡아야한다는 마음에 스스로를 채찍질했었습니다.

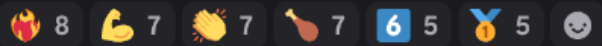
채찍질한만큼 구력에 비해 실력이 많이 늘었던 것 같고 소소하게 좋은 결과들?도 얻을 수 있었지만 솔직히 정신적으로 많이 피폐해졌습니다. 단순히 책상에 앉아있던 시간이 많아서가 아니라 남들과 비교하면서 현재 제 상황을 비관적으로 바라봤던 탓이었던 것 같습니다. 주변사람들을 보며 항상 제 공부 방향을 의심했고 이것이 목표를 위한 최단경로가 맞는지 계속 생각했던 것 같습니다.

하지만, 최근 주변 사람들과 대화를 나누면서 얻은 결론은 최단경로를 찾으려는 것 자체가 무의미하다... 입니다. 사람의 성향, 혹은 놓여져있는 상황에 따라 맞는 방향이 있을 것인데 너무 단편적으로만 생각했던 것 같습니다.

그래서 이제는 마음을 편하게 가다듬고 느리더라도 제 상황에 놓인 우선순위에 맞게 차근차근 하나씩 해나가려고합니다. 저는 오래버티고 꾸준한 건 자신있기 때문에 조금 오래걸리더라도 괜찮을 것 같습니다.

모두들 밥 잘 먹고 잘 자고 운동도 열심히 하시길 바랍니다.

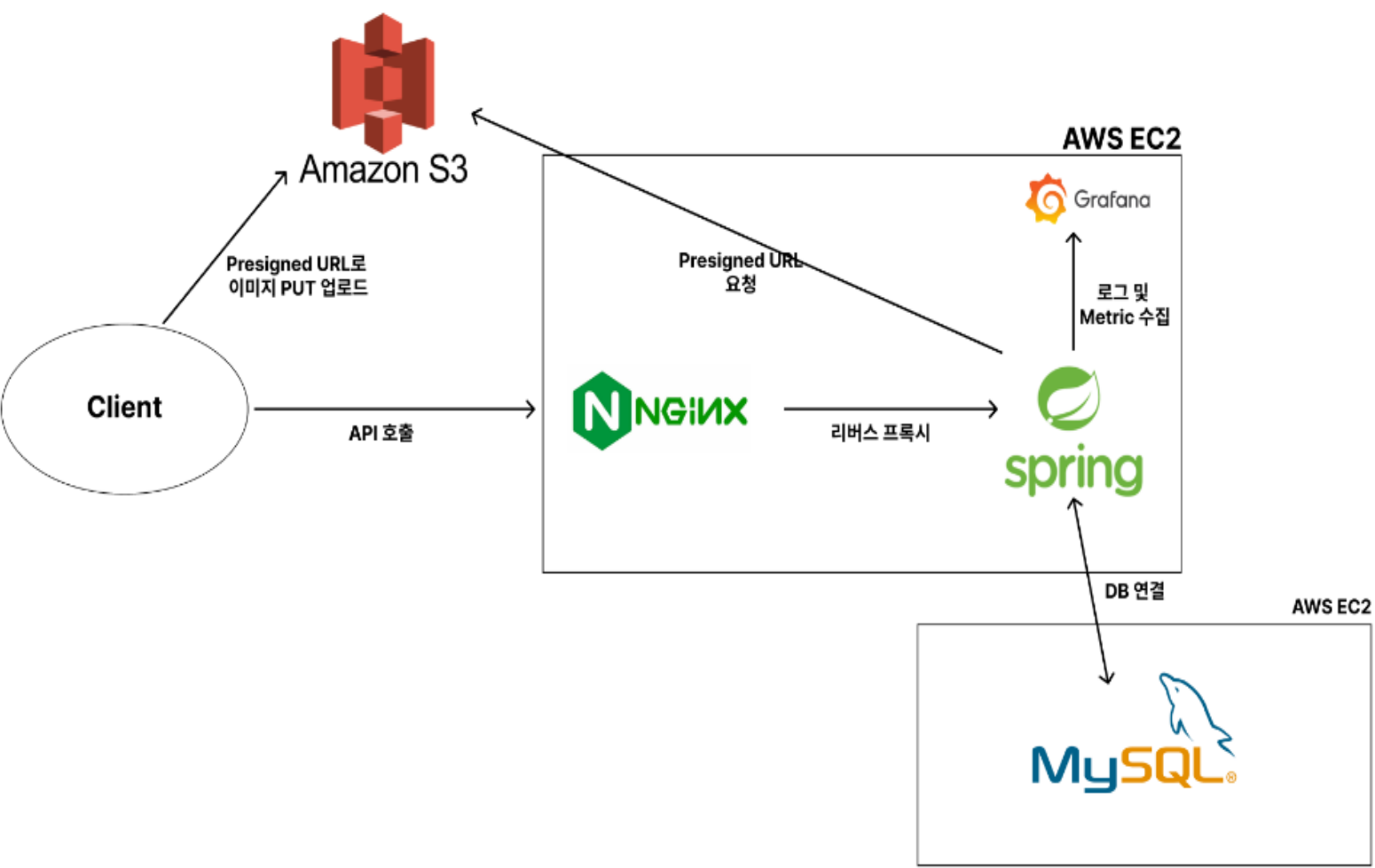
화이팅! 6days는 화이팅x6



2. 시스템 상세설계

가. 백엔드 구성도, 시스템 내 사용기술 스택 정리

‘코드몬스터’는 프론트와 백엔드, 인프라 환경이 명확하게 분리된 구조를 가지고 있으며, 안정성과 확장성을 고려한 설계를 바탕으로 운영되고 있습니다. 사용자는 웹 브라우저를 통해 React.js 기반의 프론트엔드에 접근하며, 이후 요청은 Spring Boot 기반의 백엔드 서버로 전달됩니다. 이때 모든 데이터는 MySQL 데이터베이스에 저장됩니다. 또한 로그 기반의 분석과 운영 관리를 위해 Grafana를 활용한 시각화 환경을 구축하였고, 로그 수집 도구인 Prometheus와 Loki를 통해 사용자 행동 데이터를 실시간으로 수집하고 있습니다. 이를 통해 플랫폼 성능 모니터링 뿐만 아니라 사용자 피드백 기반의 기능 개선에도 기여하고 있습니다.



● 사용 기술 스택 정리

영역	사용 기술
프론트엔드	React.js
백엔드	Spring Boot, Spring Security, Spring JPA
데이터베이스	MySQL
로그 수집 및 시각화	Loki, Prometheus, Grafana
서버 및 배포	AWS EC2, Nginx
스토리지	AWS S3

3. 개발결과

가. 서비스 이용의 흐름도, 기능설명, 새롭게 개선한 기능

‘코드몬스터’는 사용자 회원가입부터 스터디 참여, 문제 추천, 풀이 기록, 회고 작성, 피드백 수집까지 이어지는 전반적인 학습 사이클을 하나의 플랫폼 내에서 경험할 수 있도록 설계되었습니다. 전체적인 서비스 이용 흐름은 다음과 같습니다.

1. 회원가입 및 로그인

- 사용자는 카카오톡 기반의 소셜 회원가입 후 로그인합니다.
- 디스코드에 참여하여 학습 목적, 코딩테스트 경험 수준 등을 바탕으로 적절한 스터디 팀에 배정됩니다.

2. 스터디 참여 및 문제 추천

- 각 팀은 요일별 학습 스케줄을 기반으로 알고리즘 문제를 추천받습니다.
- 추천된 문제는 난이도, 주제 분류, 기존 풀이 여부 등을 고려하여 자동으로 선택됩니다.

3. 문제 풀이 기록 및 회고 작성

- 사용자는 각 문제에 대한 풀이 과정을 작성하고, 팀원들과 공유합니다.
- 회고 기능을 통해 본인의 학습 내용을 정리하고 팀원들과 피드백을 주고 받을 수 있습니다.

4. 행동 데이터 수집 및 서비스 개선

- API 요청 기록, 페이지 이동 로그, 스터디 참여 빈도 등 사용자 데이터를 수집합니다.
- 수집된 데이터는 트래픽 분석, 병목 파악, 맞춤형 기능 개선 등에 활용됩니다.

1학기 중간보고서 이전 기간 동안, 저희는 이미지 업로드 기능의 구조적 한계를 극복하고 사용자 경험을 개선하기 위한 기능 고도화 작업을 진행하였습니다. 기존에는 사용자가 문제 풀이 글을 작성할 때 이미지 파일을 백엔드 서버로 전송하고, 서버가

이를 AWS S3에 직접 업로드하는 방식이었습니다. 이 방식은 초기 개발 단계에서는 구현이 간단하다는 장점이 있었지만, 실제 서비스 운영 과정에서 다음과 같은 여러 문제점이 발생하였습니다.

첫째, 게시글 수정 시 기존 이미지와 새 이미지가 서로 다른 방식으로 전달되어 이미지 순서를 일관되게 유지하기 어려웠습니다. 이를 해결하기 위해 프론트엔드에 복잡한 트래킹 로직을 추가해야 했고, 유지보수가 어려워지는 문제가 뒤따랐습니다. 둘째, 이미지 업로드 속도 저하 문제가 발생하였습니다. 이미지 파일이 백엔드 서버를 거쳐야만 S3에 저장될 수 있었기에 큰 이미지가 첨부되는 경우 네트워크 지연이 빈번히 발생하였고, 이는 사용자 이탈로 이어질 수 있는 위험 요소였습니다. 셋째, 백엔드 서버가 이미지 바이트 데이터를 직접 처리하면서, CPU, 메모리, I/O 리소스 사용량이 증가해, 다른 비즈니스 로직의 성능에도 부정적인 영향을 주기 시작하였습니다.

이러한 문제들을 해결하기 위해 저희는 AWS S3의 Presigned URL 기능을 도입하였습니다. 사용자가 이미지를 업로드하면, 백엔드에서는 사전에 메타데이터(파일명, MIME 타입 등)를 기반으로 S3에 접근 가능한 URL을 발급하고, 프론트엔드는 해당 URL을 통해 직접 이미지를 업로드하는 방식으로 구조를 전면 개편하였습니다. 이로 인해 서버의 부하를 줄이고, 이미지 업로드 속도를 대폭 향상시킬 수 있었으며, 보안 측면에서도 Presigned URL의 만료 시간이 자동으로 설정됨으로써 안정적인 전송이 가능해졌습니다.

실제 개선 결과, 3개의 이미지를 포함한 게시글 작성 API의 평균 응답 시간이 기존 236ms에서 18ms로 줄어드는 성과를 얻었습니다. 이는 약 10배 이상의 속도 개선이며, 사용자 만족도에도 긍정적인 영향을 미쳤습니다. 또한, 이 과정에서 운영중이던 이미지 메타데이터 테이블을 제거하고, 본문 내에 S3의 Public URL을 직접 포함시키는 방식으로 데이터 구조를 단순화하였습니다. 이를 위해 모든 기존 게시글의 본문을 일괄 업데이트하는 마이그레이션 컴포넌트를 별도로 작성하였으며, 안정적으로 데이터를 변환하는 데 성공하였습니다.

이번 개선 작업은 단순한 기능 최적화에 그치지 않고, 서비스 아키텍처를 보다 가볍고 명확하게 정리하는 계기가 되었으며, 프론트엔드와 협업 과정 속에서 시스템 전반에 대한 이해도와 소통 방식도 함께 성장할 수 있던 값진 경험이었습니다.

이번 작업의 자세한 구현 과정은 아래 블로그글에 자세하게 정리해두었습니다!

[블로그 링크 바로가기](#)

이번 중간보고서 이후에는 서비스의 안정성과 성능을 한층 더 끌어올리기 위해 로깅 및 모니터링 환경 구축에 집중하였습니다. 이는 단순히 새로운 기능을 개발하는 것을 넘어, 실제 서비스 운영 데이터를 기반으로 성능 병목 구간을 정확히 파악하고 선제적으로 개선하기 위함입니다. 구체적으로 API 사용량, CPU 사용량, 힙 메모리 사용량 등 핵심 지표들을 실시간으로 모니터링하여 기능 개선이 필요한 부분을 신속하게 식별하고 관리할 수 있는 기반을 마련하였습니다. 이를 위해 크게 두 가지 환경을 구축하였습니다.

첫 번째로 메트릭(Metric) 기반의 모니터링 설정입니다. 저희는 서비스의 안정적인 운영과 선제적인 성능 개선을 위해 Prometheus와 Grafana를 활용한 매트릭 기반 모니터링 시스템을 구축하였습니다. 이 시스템은 Spring Boot 애플리케이션에서 발생하는 다양한 성능 지표를 수집, 저장, 시각화하여 서비스의 현재 상태를 실시간으로 파악하고 문제 발생 시 빠르게 대응할 수 있도록 돕습니다.

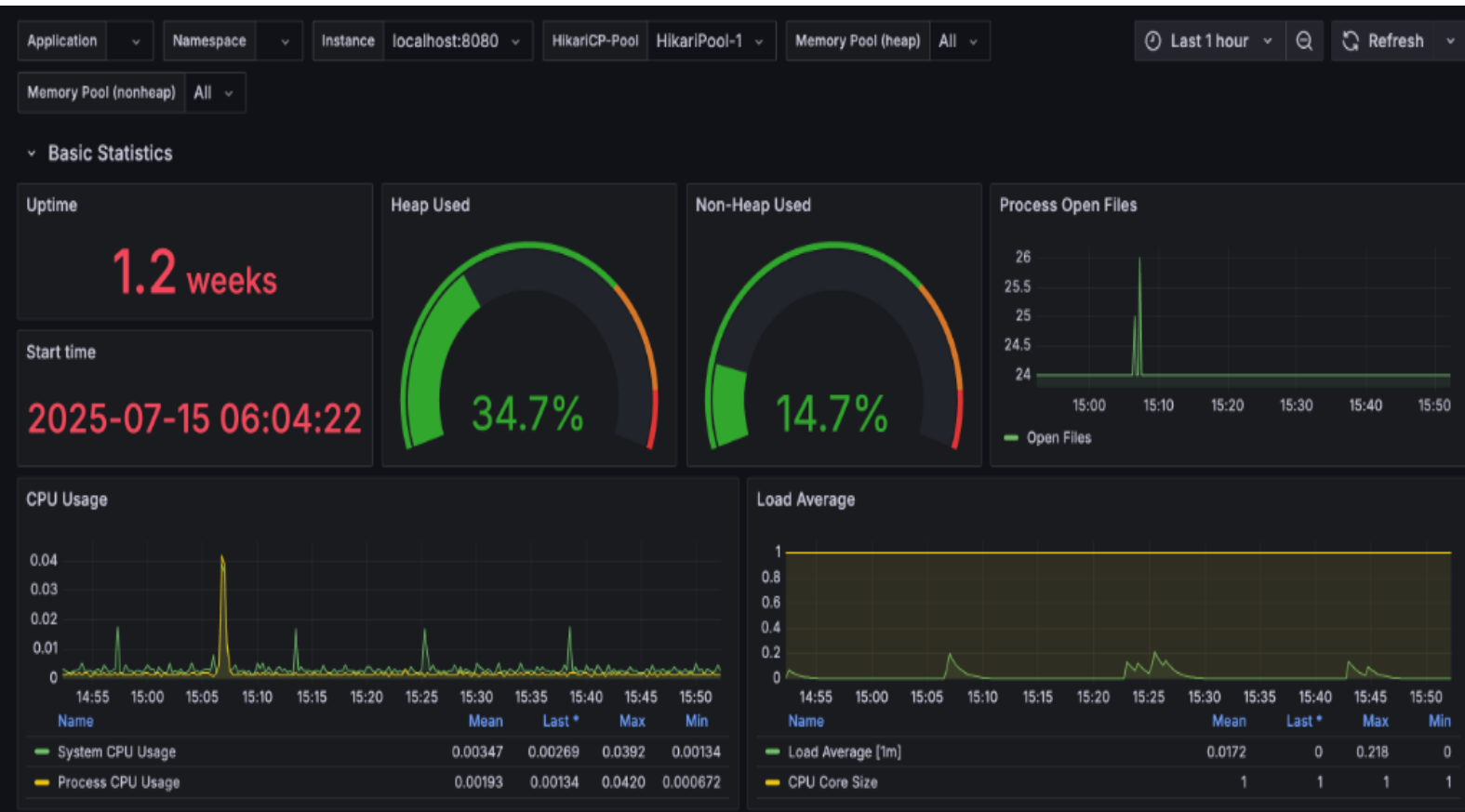
전체적인 모니터링 흐름은 다음과 같습니다.

(1) **Spring Boot 애플리케이션** : `spring-boot-starter-actuator`를 통해 JVM 메모리 사용량, HTTP 요청 수, 응답 시간, CPU 사용률, DB 커넥션 수 등 Micrometer 기반의 메트릭을 Prometheus 포맷으로 노출하도록 설정하였습니다.

(2) **Prometheus** : 앞서 설정한 `/actuator/prometheus` 엔드포인트로부터 15초마다 주기적으로 메트릭 데이터를 스크랩하여 시계열 메트릭 DB에 저장하도록 구성하였습니다. 이를 통해 시스템의 과거 및 현재 성능 데이터를 효과적으로 기록할 수 있게 되었습니다.

(3) **Grafana** : 수집된 메트릭 데이터를 시각화하기 위해 EC2에 Grafana를 설치하였습니다. Nginx 리버스 프록시를 통해 특정 서브 경로로 Grafana 웹 UI에 접근할 수 있도록 설정하고, Grafana 파일 또한 서브 경로 동작을 지원하도록 변경하였습니다. 최종적으로 Grafana에 Prometheus를 데이터 소스로 등록하고, Spring Boot 애플리케이션의 주요 메트릭을 한눈에 파악할 수 있는 대시보드를 만들어 시각화 환경을 아

래와 같이 완성하였습니다!



두 번째로 저희는 서비스에서 발생하는 애플리케이션 로그를 효과적으로 수집하고 분석하여 문제 해결 시간을 단축하기 위해 로그 기반 모니터링 시스템을 구축하였습니다. 이 시스템은 Promtail, Loki, Grafana의 조합으로 이루어져 있으며, Loki 공식 문서를 참고하여 구현하였습니다.

전반적인 로그 모니터링 흐름은 아래와 같습니다.

- (1) **Spring Boot 애플리케이션** : logback을 통해 콘솔과 파일에 로그를 출력하도록 설정하여, 모든 애플리케이션 이벤트를 기록할 수 있도록 하였습니다.
- (2) **Promtail (로그 수집기)** : EC2 인스턴스에 설치된 Promtail은 Spring Boot 애플리케이션의 로그 파일을 실시간으로 감시합니다. 각 로그 라인에는 라벨을 부여하여 로그를 분류하고, 이 데이터를 Loki의 HTTP API로 푸시합니다.
- (3) **Loki (로그 저장소)** : Promtail로부터 전송받은 로그 데이터를 시계열 형태로 저장하는 역할을 합니다. Loki는 전통적인 로그 시스템과 달리 별도의 인덱싱 없이 라

벨 기반으로 로그를 저장하여 빠른 검색 및 조회가 가능하며, Grafana의 요청에 따라 저장된 로그를 응답으로 제공합니다.

(4) **Grafana (대시보드 시각화)** : Grafana 웹 UI에 Loki를 새로운 데이터 소스로 등록하여 로그를 시각화합니다. Grafana의 Explore 탭이나 커스터마이징 가능한 대시보드 패널을 통해 LogQL 쿼리를 사용하여 특정 시간대, 요청 ID, 에러 메시지 등을 다양한 조건으로 손쉽게 조회하고 필터링할 수 있습니다.

이러한 로그 기반 모니터링 시스템 구축을 통해, 개발팀은 클라이언트에 반환된 에러 응답이 로그에 기록되지 않는 문제와 같이 복잡한 상황에서도 특정 경로에서의 에러 발생을 정확하게 감지하고 추적할 수 있게 되었습니다.

```
> 2025-07-20 19:59:53.124
> 2025-07-20 19:59:53.124      at java.base/java.lang.Thread.run(Thread.java:1583) ~[na:na]
> 2025-07-20 19:59:53.124      at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.124      at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.124      at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1190) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.124      at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.124      at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1741) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.115      at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:985) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.115      at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.115      at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:257) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.115      at org.apache.coyote.http11.Http11InputBuffer.parseRequestLine(Http11InputBuffer.java:487) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 19:59:53.114 java.lang.IllegalArgumentException: Invalid character found in method name [0x040x010x00P0xd0_p0x010x00...]. HTTP method names must be tokens
> 2025-07-20 19:59:53.111
> 2025-07-20 19:59:53.110 Note: further occurrences of HTTP request parsing errors will be logged at DEBUG level.
> 2025-07-20 19:59:53.110 2025-07-20T19:59:52.867+09:00 INFO 1234450 --- [Backend] [http-nio-8080-exec-9] o.apache.coyote.http11.Http11Processor : Error parsing HTTP request header
> 2025-07-20 12:47:55.387
> 2025-07-20 12:47:55.387      at java.base/java.lang.Thread.run(Thread.java:1583) ~[na:na]
> 2025-07-20 12:47:55.387      at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 12:47:55.387      at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) ~[tomcat-embed-core-10.1.30.jar!/:na]
> 2025-07-20 12:47:55.387      at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1190) ~[tomcat-embed-core-10.1.30.jar!/:na]
```

4. 서비스 홍보 및 이용방법

저희 서비스는 웹 기반 서비스로 별도의 설치 없이 누구나 접속할 수 있도록 개발되었습니다. 사용자는 [코드몬스터 사이트](#) 링크를 통해 사이트에 접속한 뒤, 카카오톡 계정으로 간편히 가입할 수 있습니다.

초기 사용자 유입을 위해 팀의 PM과 마케터는 대학 커뮤니티(에브리타임), 인프런 게시판 등을 통해 활발한 홍보 활동을 진행하였습니다. 이와 같은 커뮤니티 기반의 유입 전략을 통해 단기간 내 100명 이상의 사용자들을 확보할 수 있었습니다.

이외에도 노션을 통해 사용자 온보딩 가이드를 제공하고 있으며, 사이트 및 디스코드 사용법, 팀 선택 방법, 코드 작성 및 기록 방식, 디스코드 활용 예시 등을 체계적으로 정리하였습니다. 이를 통해 신규 사용자들이 큰 어려움 없이 스터디에 빠르게 적응하고, 꾸준한 학습을 이어갈 수 있도록 지원하고 있습니다.

코드몬스터 온보딩 가이드 : [사이트 바로가기](#)

코드몬스터 인프런 게시판 홍보글 : [사이트 바로가기](#)

5. 팀 소개 (소속, 구성원별 역할)

(1) 김태윤 - 정보컴퓨터공학부 소속, 팀장 겸 백엔드 개발자

프로젝트의 전반적인 진행을 이끌며 팀장을 맡았습니다. 백엔드 개발을 담당하며 서비스의 기능 고도화 및 효율적인 아키텍처 구현에 집중하였습니다. 특히, 이미지 업로드 성능 개선과 같은 주요 백엔드 로직 개발 및 보수에 힘 썼습니다.

(2) 장진영 - 정보컴퓨터공학부 소속, 백엔드 개발자

백엔드 시스템의 핵심 기능 개선 및 아키텍처 설계를 주교하고, 안정적인 서비스 운영을 위한 로깅 및 모니터링 시스템 구축에 크게 기여하였습니다.

(3) 이민경 - 정보컴퓨터공학부 소속, 프론트엔드 개발자

개발된 모든 기능이 사용자에게 효과적으로 전달될 수 있도록 웹사이트에 직접 적용하고 구현하는 역할을 수행하였습니다. 특히 랜더링 최적화 개선에 힘 썼습니다.

(4) 서혜성 - 디자인학과 소속, UI/UX 디자이너

전반적인 UI 및 UX 디자인을 책임졌습니다. 사용자 친화적인 레이아웃과 직관적인 디자인을 통해 학습 효율성을 높이는 데 주력하였습니다.

(5) 김민욱 - 미디어커뮤니케이션학과, 마케터

‘코드몬스터’ 플랫폼 확산을 위해 SEO 전략 수립 및 실행, 인스타그램을 통한 홍보 활동 등 마케팅 전반을 담당하였습니다. (인스타그램 아이디 : codemonster_official)

6. 참여후기

이번 SW학습공동체 활동에 참여하며 단순한 개발을 넘어 사용자와 함께 성장하는 특별한 경험을 하였습니다. 무엇보다 이번 기회를 통해 저희의 개발 목표를 다시 한번 명확히 설정할 수 있었습니다. 특히 이미지 업로드 기능의 비효율적인 구조를 개선하고 AWS S3 Presigned URL을 도입하며 평균 응답 시간을 10배 이상 단축시키는 등 실제 사용자들이 체감할 수 있는 긍정적인 변화를 만들어냈습니다. 이는 단순히 기능을 추가하는 것을 넘어, 서비스 아키텍처를 가볍고 명확하게 정리하는 귀중한 경험이었습니다.

또한, 안정적인 서비스 운영을 위한 시스템 구축에도 집중하였습니다. Prometheus, Loki, Grafana를 활용한 메트릭 및 로그 기반 모니터링 시스템을 성공적으로 도입하며, 트래픽 병목 현상이나 잠재적인 성능 이슈를 사전에 감지하고 해결할 수 있는 기반을 마련하였습니다. 이러한 작업들은 저희의 기술적인 역량을 한 단계 끌어올리는 중요한 발판이 되었습니다.

가장 큰 수확은 팀원 간의 화합과 동기 부여를 더욱 단단히 만들었다는 점입니다. '여름-인사방' 채널을 통해 코드몬스터 참여자들과 직접 소통하며 그들의 학습 고민과 성공 사례를 들을 수 있었던 점은 큰 울림으로 다가왔습니다. 사용자들의 솔직한 이야기들은 저희가 어떤 가치를 제공해야 하는지 다시금 일깨워줬고, 팀원 모두가 '우리가 만든 서비스가 누군가에게 실질적인 도움을 주고 있구나'하는 보람을 느끼게 하였습니다.

이번 활동으로 '코드몬스터'는 단순한 코딩 테스트 준비 플랫폼을 넘어, 개발자와 학습자들이 서로 지지하고 함께 성장하는 살아있는 커뮤니티로 발전할 가능성을 보았습니다! 저희는 이 경험을 토대로 앞으로 더욱 견고하고 사용자 친화적인 서비스를 만들어 나갈 계획입니다.