

PNU SW학습공동체 최종(중간)보고서

Github로 파일업로드하여 제출

1. 프로젝트 소개

가. 배경 및 필요성

개발자들은 코딩테스트 준비에 큰 부담을 느끼며 비효율적인 방식으로 스터디를 꾸준히 진행하지 못한다는 현실을 목격하였습니다. 특히 20명 이상의 개발자 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

‘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 100여 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.

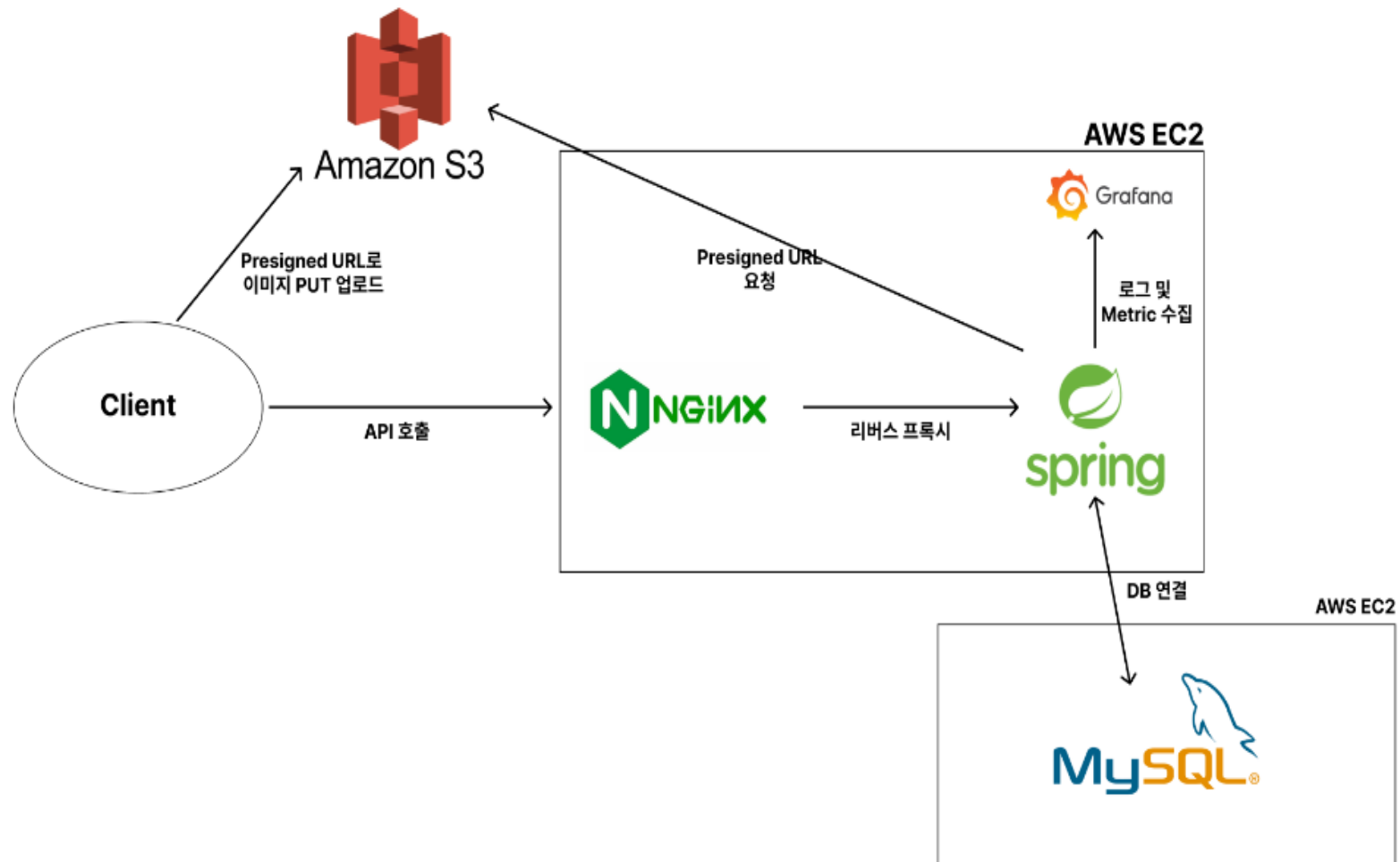
다. 사회적가치 도입 및 추후 계획

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 제공받을 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.

2. 시스템 상세설계

가. 백엔드 구성도, 시스템 내 사용기술 스택 정리



‘코드몬스터’는 프론트와 백엔드, 인프라 환경이 명확하게 분리된 구조를 가지고 있으며, 안정성과 확정성을 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 지정됩니다.

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](#)

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 노선을 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](#)