

# fair.gg

전적 검색, 분쟁 조정, 팀원 찾기를 하나로

팀원: 유진호

발표일: 2025. 09. 10.



# 프로젝트 요약 및 핵심 가치

## 프로젝트 요약

OP.GG와 같은 LoL 전적 검색을 기반으로, 커뮤니티 기능을 결합한 웹 서비스입니다.

주요 기능은 '**분쟁 조정**'과 신뢰도 기반 '**팀원 찾기**'입니다.

## 핵심 가치

객관화된 분쟁 판정과 신뢰 가능한 팀원 매칭으로 건강하고 지속가능한 플레이 환경을 만드는 것을 목표로 합니다.





## 프로젝트 배경 및 문제 제기

### 게임 트렌드의 변화

최근 LoL 플레이어는 솔로 랭크보다 친구와 함께하는 다인 게임의 비중이 증가하고 있습니다.

이로 인해 개인 실력만큼이나 팀원 간의 협력과 의사소통이 중요해졌습니다.

### 문제 상황: 끊이지 않는 인게임 갈등

"누구의 잘못인가?", "더 나은 선택은 무엇이었나?"와 같은 판단충돌이 빈번하게 발생합니다.

실제로 2023년경, BJ에게 돈을 지불하고 판결을 받는 '롤문철' 콘텐츠가 성행했을 만큼 갈등 중재에 대한 수요가 높다는 것을 확인했습니다.

하지만 '롤문철'은 도박성 문제로 제재를 받았고, 현재 유저들의 갈등을 해소할 마땅한 플랫폼이 부재한 상황입니다.

# 기존 서비스의 한계

## 기존 전적 검색 사이트

(OP.GG, FOW.KR 등)

- 전적, 챔피언 통계 등 데이터 제공에만 집중
- 게임 내 특정 장면에 대한 잘잘못을 따지거나 사후 피드백을 제공하는 갈등 중재 기능이 없음



## 기존 팀원 찾기 기능의 문제

- OP.GG의 팀원 찾기 기능은 다른 플레이어의 신뢰도를 파악할 요소가 부족
- 단순히 트롤 플레이어를 신고하는 기능만 존재
- 좋은 팀원을 변별할 장치가 미흡



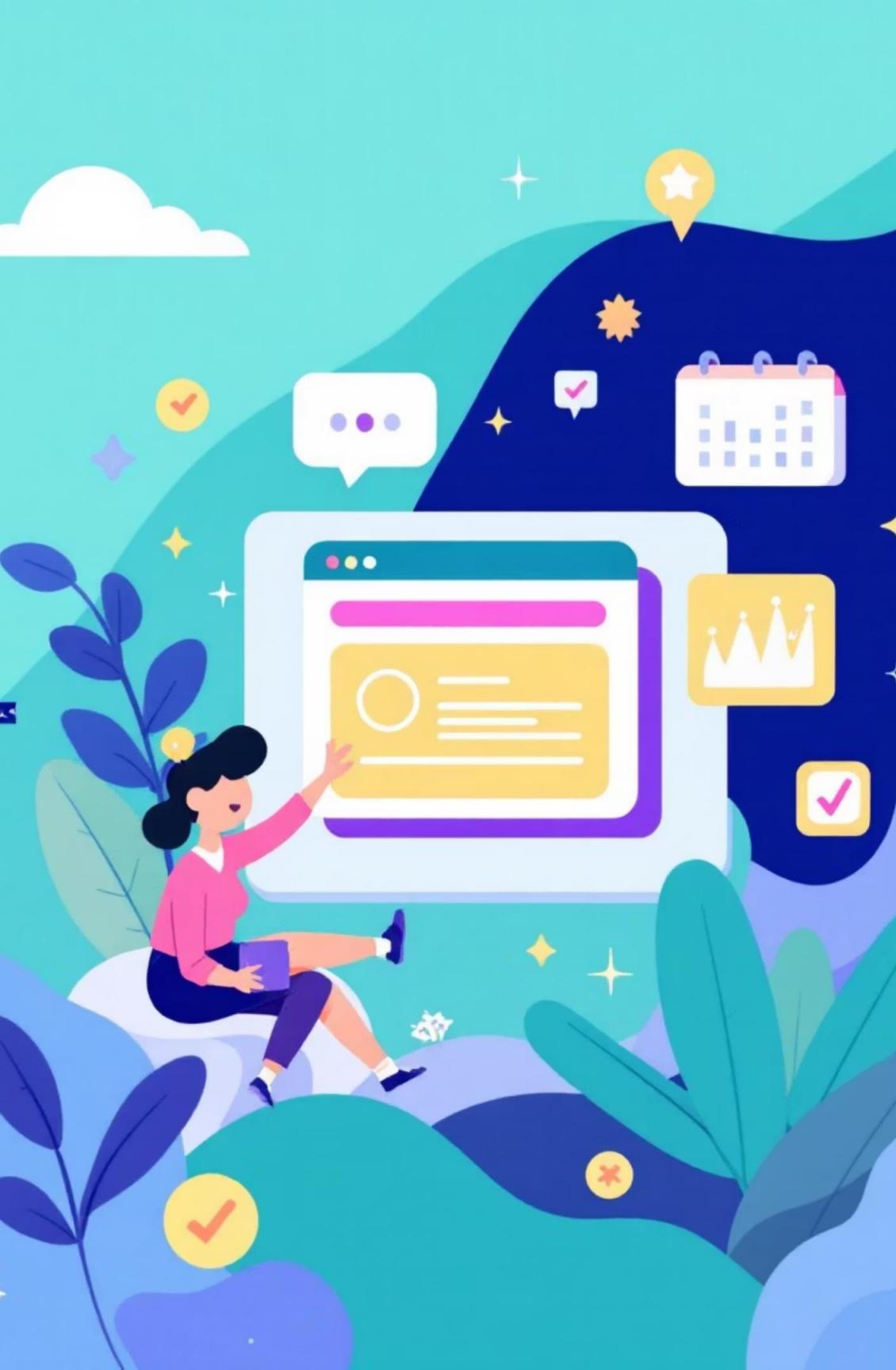
# 우리의 해결책: fair.gg

## 세 가지 핵심 기능의 선순환 구조

저희는 전적 검색, 분쟁 조정, 팀원 찾기(리뷰/신뢰도) 기능을 유기적으로 연결하여 문제 해결을 제안합니다.



이 선순환 구조를 통해 판정의 질과 매칭의 질을 동시에 개선하고자 합니다.



# 주요 기능 (1) - 분쟁 조정 시스템

01

## 분쟁 등록

유저가 영상 등과 함께 분쟁 장면과 쟁점  
을 게시판에 등록합니다.

02

## 가중 투표 참여

다른 유저들이 자신의 티어, 주 라인, 챔  
피언 정보를 바탕으로 가중 투표에 참여  
하고, 근거 코멘트를 작성합니다.

03

## 결과 제시

투표 결과는 'A 유저 책임 65%, B 유저 35%'와 같이 직관적인 스코어와 권장 플레이  
요약으로 제시됩니다.

### ✓ 차별점

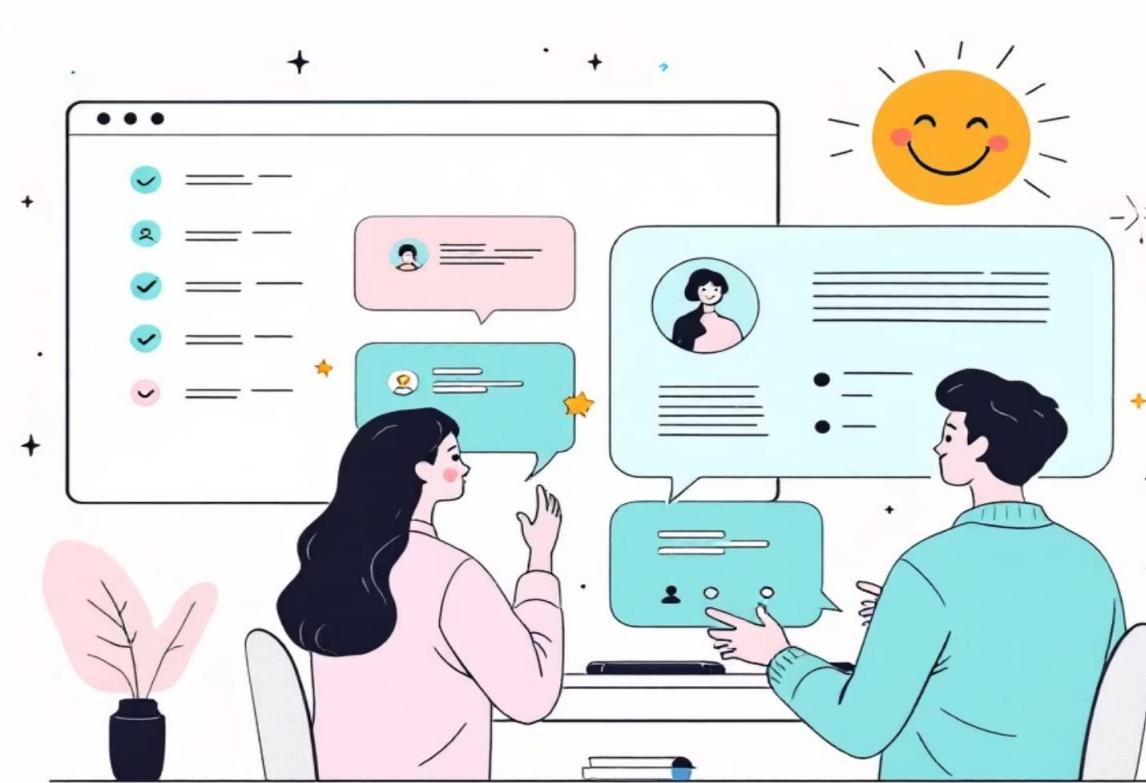
'롤문철'의 현금 요소를 배제하고, 사이트 내 활동 포인트(네이버 지식인  
의 내공과 유사)를 보상으로 제공하여 건전성을 확보했습니다.

## 주요 기능 (2) - 신뢰 기반 팀원 찾기

### 상호 리뷰 시스템

함께 게임을 한 팀원과 경기 후 상호 리뷰를 남길 수 있습니다.

리뷰 항목은 '협업 능력', '매너', '오더' 등으로 세분화됩니다.



### 신뢰도 지표

축적된 상호 리뷰와 분쟁 조정 참여 이력, 획득한 포인트/배지 등이 종합되어 각 유저의 '신뢰도 지표'가 됩니다.

팀원을 구할 때, 이 신뢰도 지표를 참고하여 단순 전적 외에 더 많은 정보를 바탕으로 좋은 팀원을 선택할 수 있습니다.



# 가중 투표(판정) 모델

목표: '전문성, 적합성, 신뢰성'을 반영하여 판정의 정확도를 높이는 것입니다.

1

## 티어 가중치

Master 이상(x3), Diamond/Platinum(x2), Gold 이하(x1)

2

## 역할 적합성

분쟁 발생 라인과 동일한 포지션 유저에게 가산점 부여 (+1)

3

## 챔피언 숙련도

해당 챔피언의 플레이 숙련도(마스터리, KDA 등)를 지표로 환산하여 반영

④

## 발전 가능성

이번 프로젝트에서는 제외하지만, 향후 축적된 데이터를 기반으로 머신러닝을 접목하여 가중치 알고리즘을 고도화할 수 있습니다.





# 기술 스택 및 아키텍처



## Front-end

Node.js, React

사용자 친화적인 UI/UX 구현 및 Riot API 데이터 시각화



## Back-end

Java, Spring Framework

안정적인 서버 환경 및 가중 투표 로직, 신뢰도 시스템 구현



## Infra & Deployment

Docker를 이용한 컨테이너화

AWS (EC2, RDS 등)를 활용한 클라우드 배포



## External API

Riot API를 기반으로 유저 정보 및 매치 데이터를 조회

# 개발 일정

Phase	기간	주요 내용
1. 준비 및 학습	1-2주차	기획 구체화, UI/UX 설계, DB설계, React/Docker/AWS 기초 학습
2. 핵심 기능 개발	3-8주차	백엔드 API(회원, 전적), 프론트엔드 UI(로그인, 전적), 분쟁조정 CRUD 개발
3. 심화 기능 및 배포	9-12주차	가중 투표/신뢰도 시스템 개발, Docker 컨테이너화, AWS 배포, 통합 테스트
4. 최종 정리	13주차	발표 자료 제작, 최종 보고서 작성, 시연 준비 및 버퍼

# **Q & A**

질의응답

들어주셔서 감사합니다.