

성장하는 개발자 우상훈 입니다.

Creative Newcomer

010.5578.5754

zlwm0000@naver.com



About me

문제를 그냥 넘기지 않는 개발자

안녕하세요. 웹 개발자 우상훈입니다.
시간을 내어 제 포트폴리오를 봐주셔서 감사합니다.

의료, 창업, 시공 등 다양한 현장에서 일하며
항상 반복되는 문제와 비효율을 마주해 왔습니다.

“이걸 사람이 매번 해야 할까?”
“왜 이렇게 복잡할까?”

이 질문에 대한 답을 코드로 만들어가는 과정에 매력을 느꼈고,
웹 개발자의 길을 선택하게 되었습니다.

Education

2025.07 ~ 2026.01

영진직업전문학교
JAVA 디지털콘텐츠 웹 개발 과정 수료

Experience

2024.05 ~ 2025.06 | 프리랜서 시공

2023.03 ~ 2024.04 | 굽굽쿠키 디저트카페 창업

2020.06 ~ 2022.12 | 여준정형외과 방사선사

License

- 방사선사 면허증 (2020)
- 운전면허 1종 보통 (2014)

Skills

웹 개발 전반에서 사용한 기술 스택입니다.

프로젝트 및 학습을 통해 실제로 적용해본 기술들입니다.

Frontend



HTML5



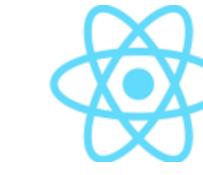
CSS3



JavaScript



TypeScript



React

웹 표준을 기반으로 화면을 구성하고
React를 활용해 컴포넌트 단위 UI를 구현했습니다.

Backend



Java



JSP



Spring Boot



Node.js

Java와 Spring Boot를 중심으로
REST API 설계 및 서버 로직을 구현했습니다.

Database



MariaDB



MySQL

관계형 데이터베이스를 사용해
테이블 설계 및 CRUD 기능을 구현했습니다.

Tools & Environment



VS Code



Eclipse



IntelliJ



Notion

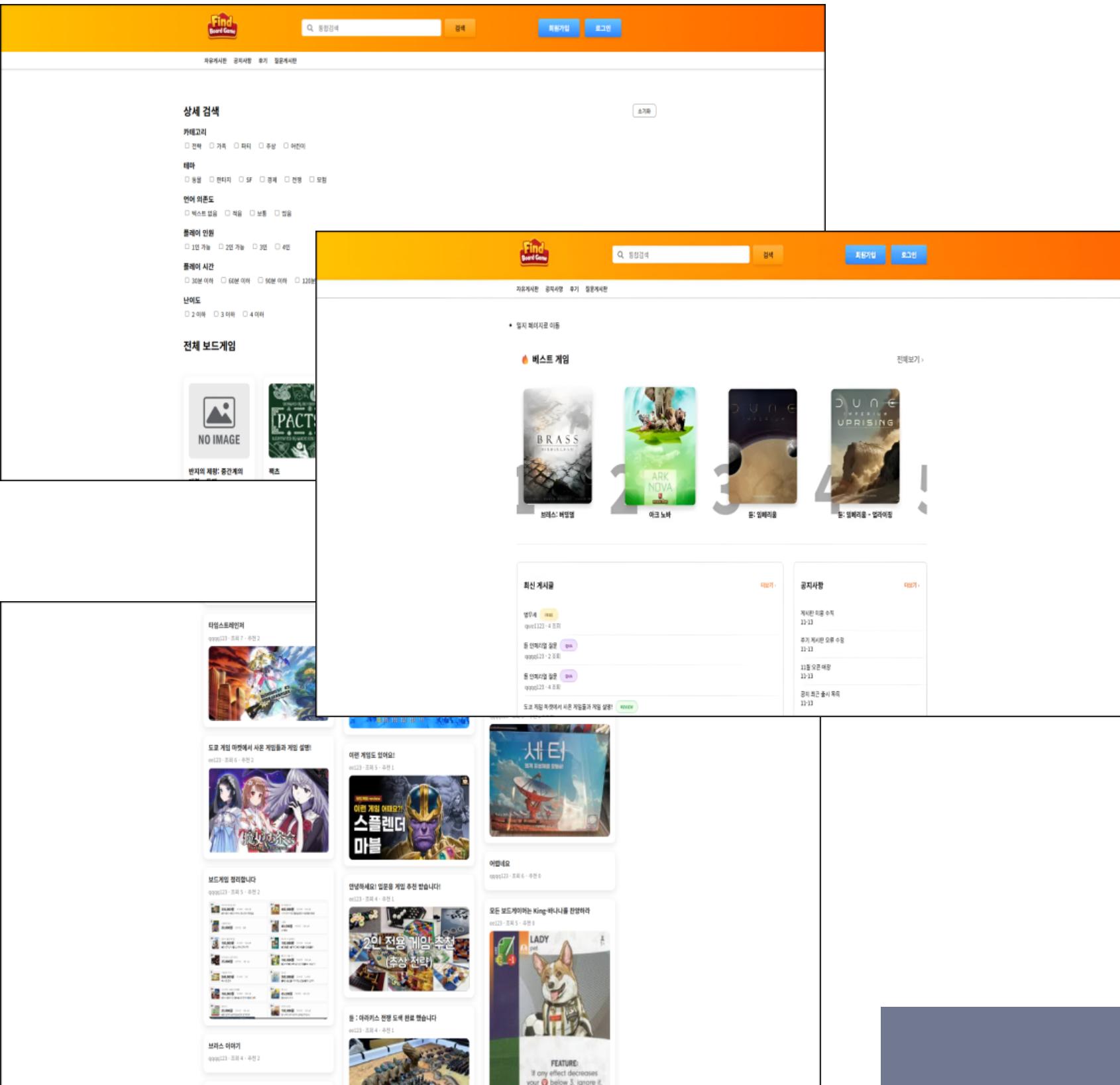


Git

Git, Notion을 활용하여 협상 관리와 팀 프로젝트를 진행하였고,
IDE 기반 개발 환경을 이용하였습니다.

Projects

FBG Find Board Game



소개

FBG는 사용자가 원하는 보드게임을 검색하고, 게임 정보를 확인하며, 게시판을 통해 다른 사용자들과 소통할 수 있는 보드게임 정보 & 커뮤니티 웹 서비스입니다.

기술 스택

Frontend



HTML5



CSS3



JavaScript

Backend



Spring Boot

Database



MariaDB

보드게임 검색 & 정보 조회

- 카테고리 별 상세 검색 기능
- 게임명 기반 검색

사용자 커뮤니티(게시판)

- 게시글 작성 / 수정 / 삭제 기능 구현
- 댓글, 대댓글 기능 구현

Projects

FBG Find Board Game

상세 검색

이벤트 기반 비동기 UI로직

클라이언트 사이드 필터링 컨트롤러로직

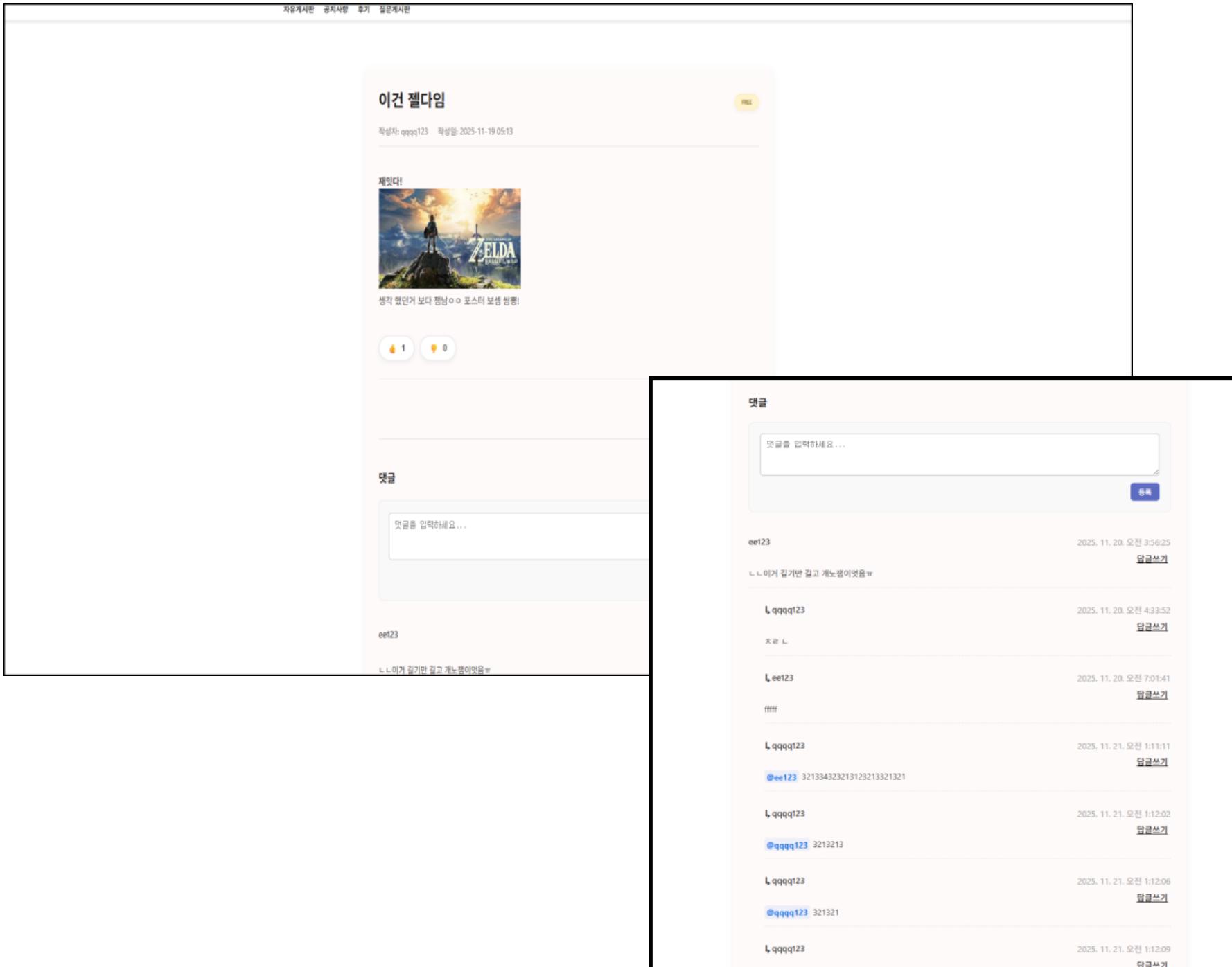
AJAX 기반 비동기 데이터 처리

DOM 조작 기반 렌더링 로직

“체크박스 이벤트를 기반으로 비동기 API를 호출하고,
결과 데이터를 DOM에 부분 렌더링하는 이벤트 기반 UI로직입니다.”

Projects

FBG Find Board Game



댓글 & 대댓글

페이지 로드

- fetchComments()
- 서버에서 댓글 JSON 조회
- renderComments()
- 댓글 → 트리 구조 변환
- renderTree()로 HTML 생성

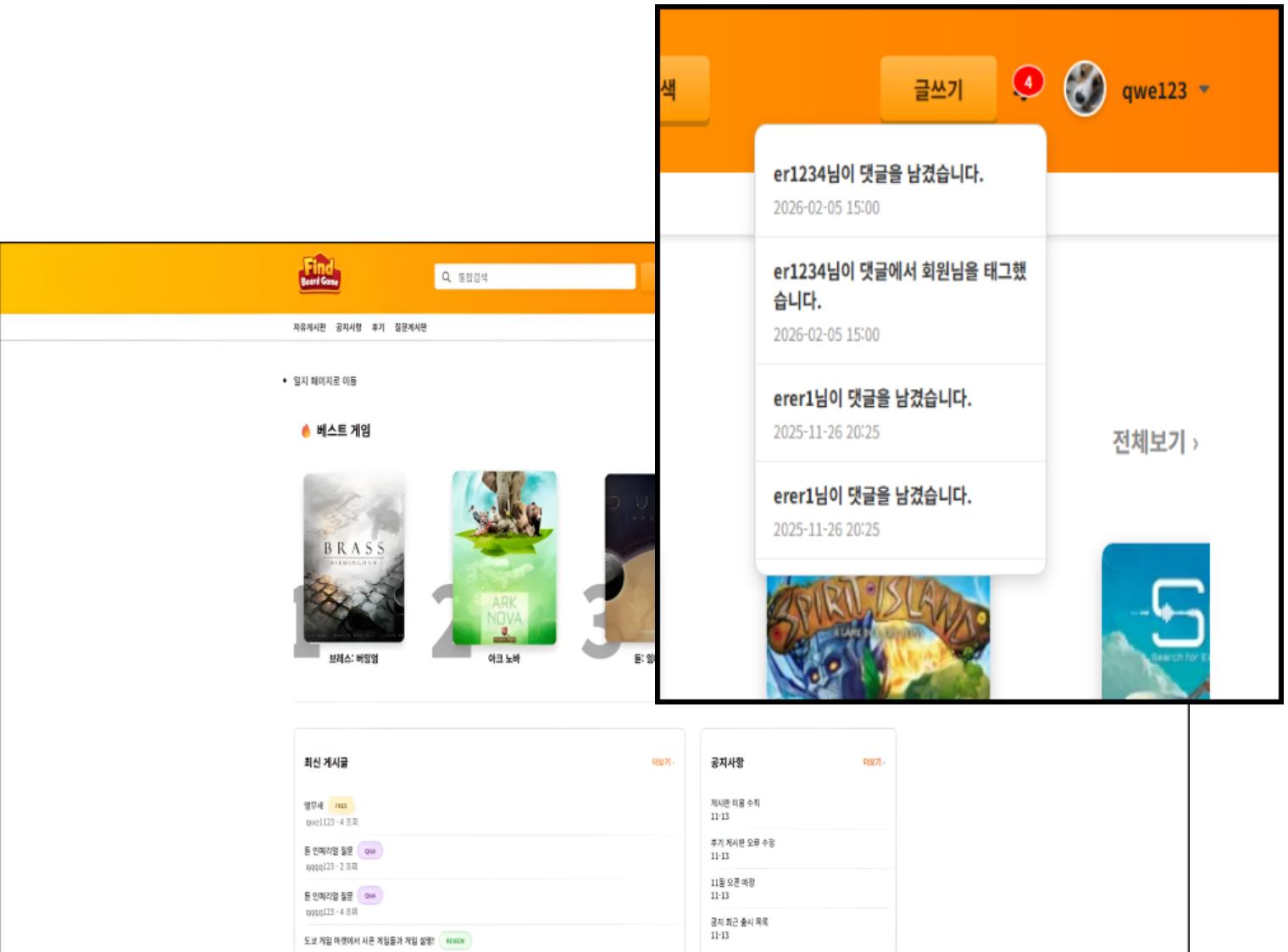
사용자 행동

- 댓글 등록
- 답글(대댓글) 등록
- 댓글 수정
- 댓글 삭제
- 좋아요 / 싫어요

"**대댓글을 포함한 트리 구조 댓글 시스템을 설계하고, 비동기 통신과 재귀 렌더링으로 UI를 구현했습니다.**"

Projects

FBG Find Board Game



알림 기능

사용자 간 상호작용이 발생했을 때 이를 알림으로 전달하도록 설계했습니다.

단순 UI 효과가 아닌,
서버에서 이벤트를 감지하고 알림 데이터를 생성
하는 구조입니다.

알림 동작 방식

게시글에 댓글이 등록되면
→ 게시글 작성자에게 댓글 알림 생성

댓글 또는 대댓글에서 사용자를 태그(@아이디)하면
→ 태그된 사용자에게 멘션 알림 생성

읽지 않은 알림 개수

최신 알림 목록 형태로 표시

사용자 행동 → 서버 이벤트 → 알림 생성
이 흐름을 명확히 분리하는 것을 목표로 했습니다.

Projects

DTD Digimon Tower Defence

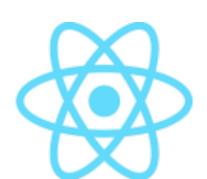


소개

DTD는 디지몬 캐릭터를 활용해 타워·몬스터·웨이브 시스템을 직접 설계하고 구현한 웹 기반 타워 디펜스 게임입니다.

기술 스택

Frontend



React



CSS3



JavaScript

Backend



Spring Boot

Database



MariaDB

게임 로직

Canvas API, requestAnimationFrame

실시간 게임 루프

- requestAnimationFrame 기반 게임 루프 구현
- 몬스터 이동 / 공격 / 사망 처리

웨이브 시스템

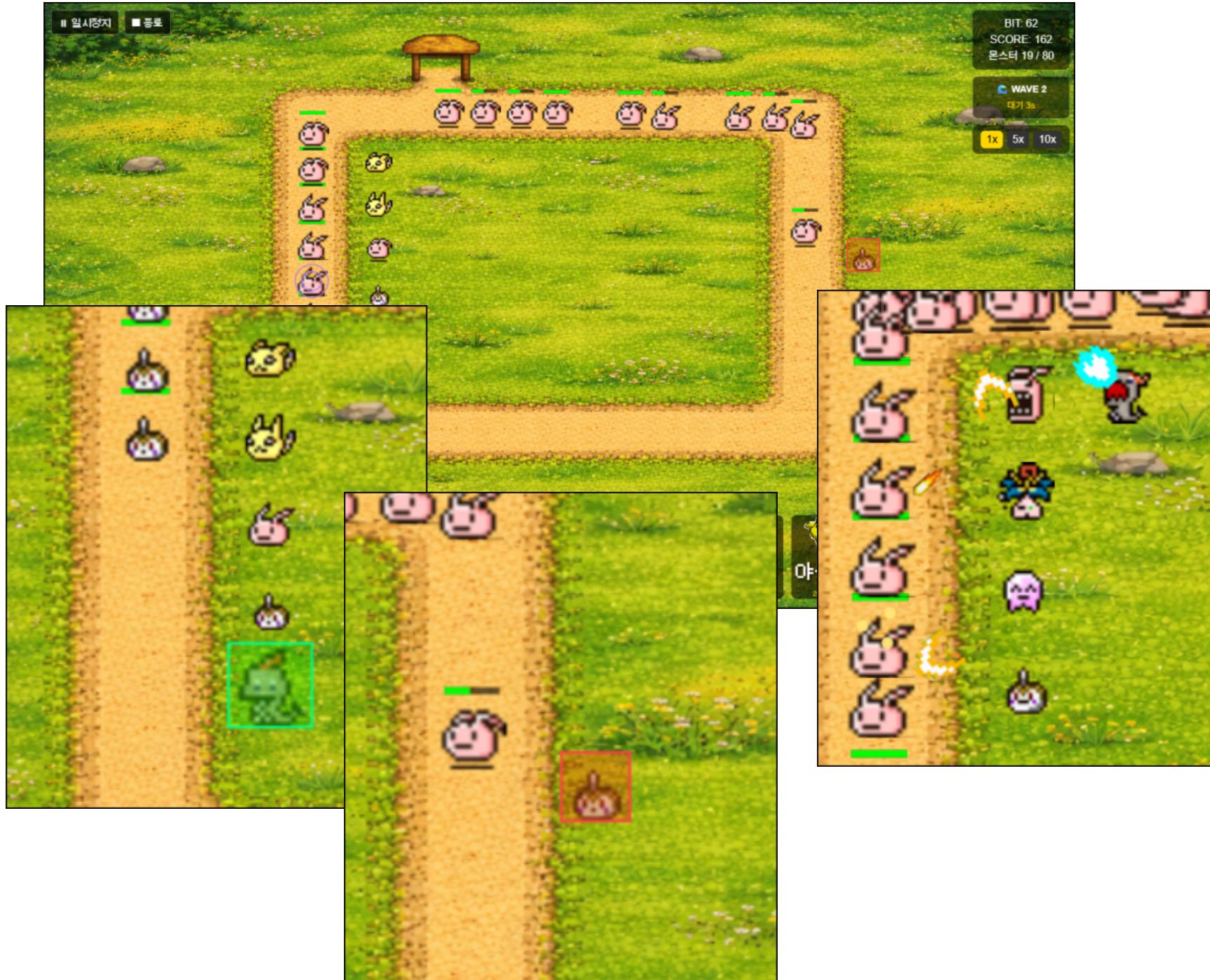
- 웨이브별 몬스터 수 및 타입 제어
- 보스 웨이브 및 미션 조건 분기 처리

타워 시스템

- 설치 가능 영역 제한
- 강화 시 능력치 실시간 반영
- 타워 철거 기능 구현

Projects

DTD Digimon Tower Defence



전체 구조

게임 시작

- 스테이지 로드 (맵, 경로, 제한 조건)
- 웨이브 시스템 시작
 - | 몬스터 생성
 - | 경로 따라 이동
 - | 타워 공격
 - | 데미지 계산
 - | 사망 / 골 도달
- 타워 설치 / 강화 / 철거
- 게임 속도 조절
- 클리어 / 실패 판정

구조

- requestAnimationFrame 기반
- 매 프레임마다 게임 상태 업데이트

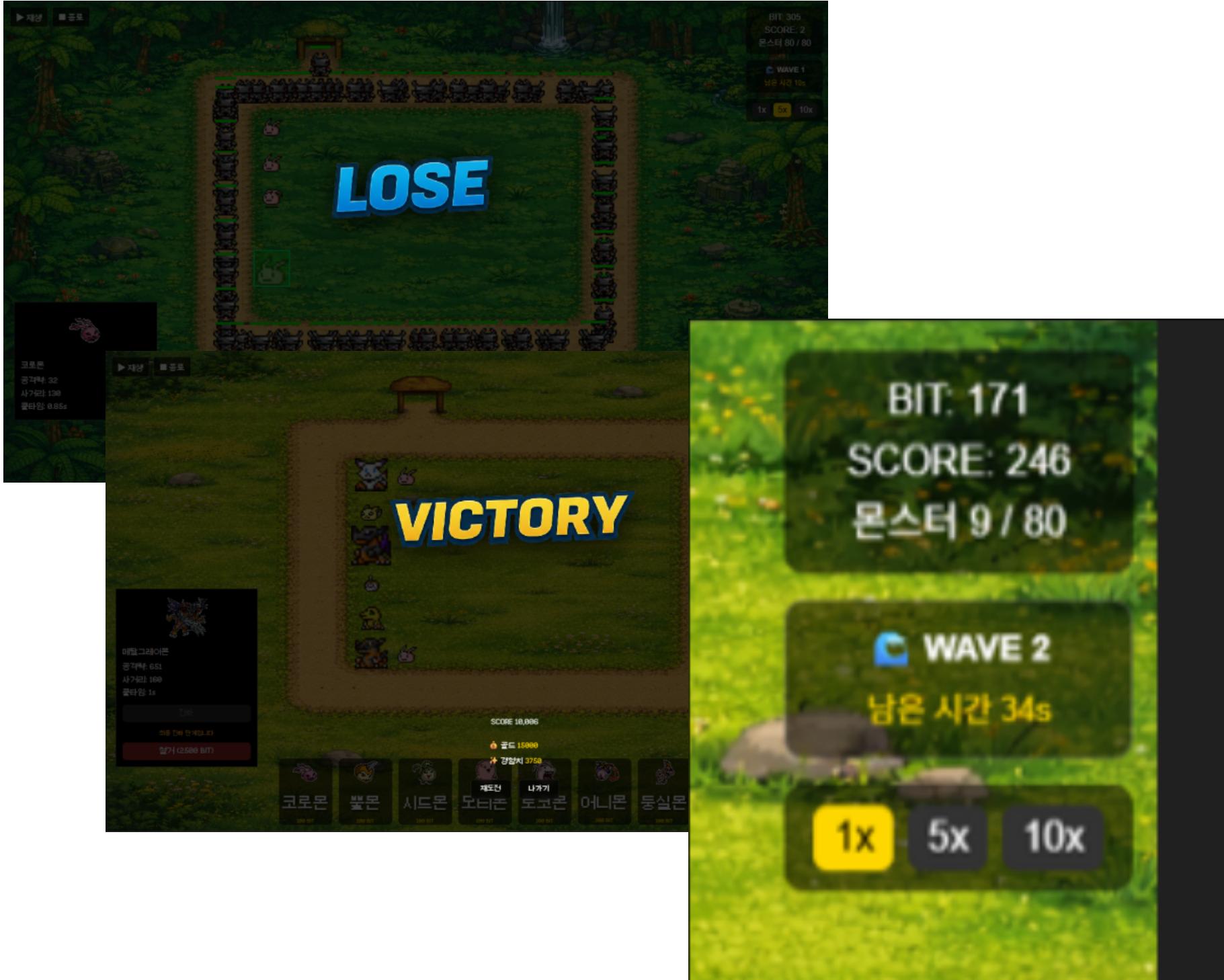
처리

- 몬스터 이동
- 타워 공격 쿨타임
- 이펙트 처리
- 웨이브 상태 체크

“버튼 클릭 → 결과”가 아니라
“시간 흐름에 따른 상태 변화”를 직접 구현 하였습니다.

Projects

DTD Digimon Tower Defence



타워 시스템, 게임 상태 관리

타워 기능

- 설치 가능 영역 제한
- 공격 범위 계산
- 강화 시 능력치 상승
- 철거 가능

핵심

- 타워 ↔ 몬스터 간 거리 계산
- 실시간 공격 대상 선택

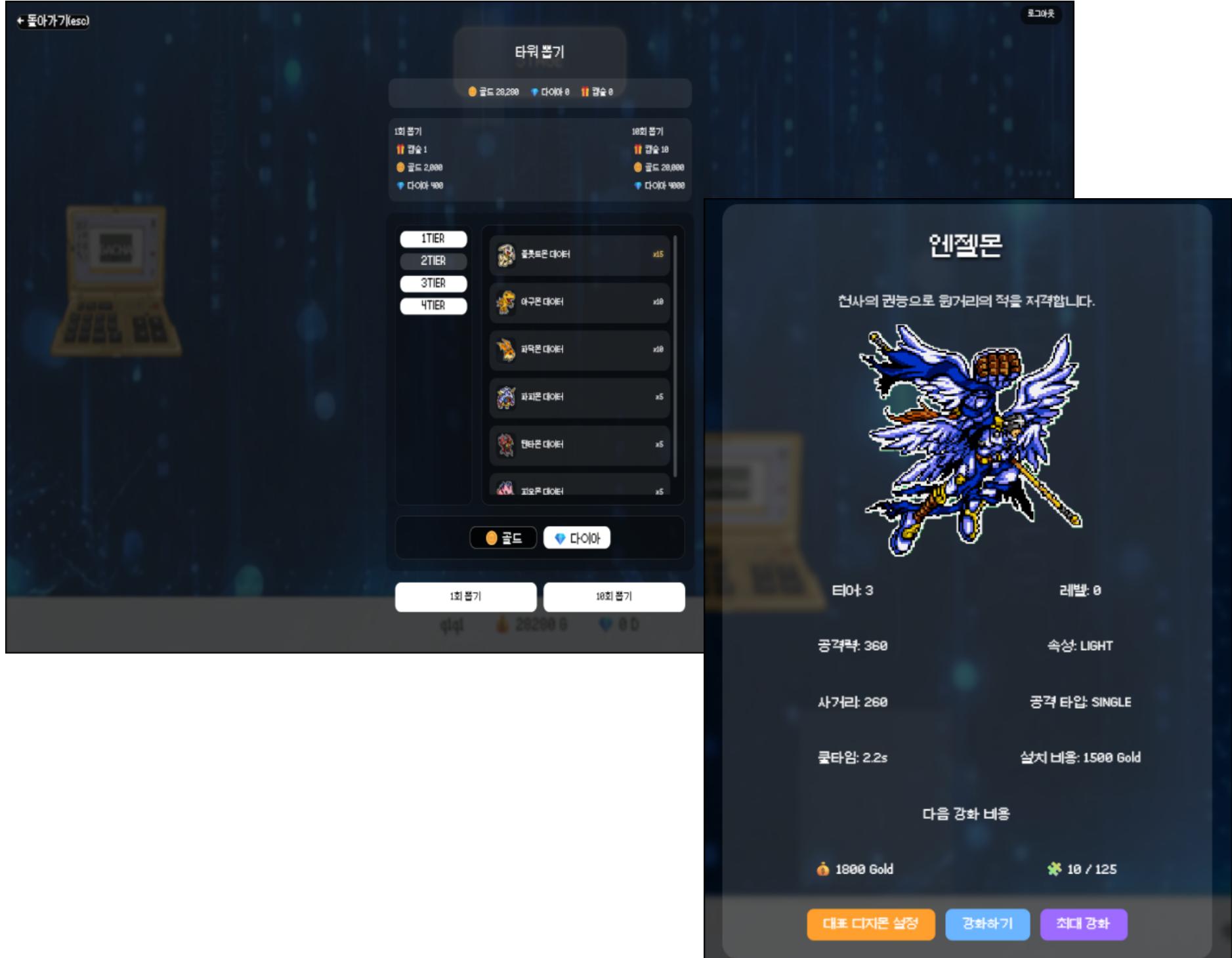
게임 상태

- 게임 속도 (배속)
- 현재 웨이브
- 남은 몬스터 수
- 클리어 / 실패 상태

“상태 하나만 꼬여도 게임이 실행되지 않았고,
구조 설계 중요성 체감할 수 있었습니다.”

Projects

DTD Digimon Tower Defence



가차, 강화 시스템

가차 시스템 구조

가차 버튼 클릭

- └ 가차 재화 소모
- └ 확률 테이블 기반 강화 재화 지급
- └ 소량 강화 재화
- └ 중간 강화 재화
- └ 고급 강화 재화

타워 진화 구조

기본 타워

- └ 조건 충족 (스테이지 / 미션 / 재화)
- └ 1차 진화 타워
- └ 추가 조건 충족
- └ 상위 진화 타워

확률 기반 강화 재화 테이블

- 재화 종류별 확률 설정
- 서버에서 결과 결정
- 어떤 결과든 강화에 사용 가능

강화 시스템과의 관계

- 전투 성능 향상
- 공격력 / 레벨 등 수치 증가
- 공격 방식 변화

Technical Choices

프로젝트에 맞게 기술을 선택하고, 역할을 분리 했습니다.

1) FBG – Spring Boot + Thymeleaf + JavaScript (Full-stack)

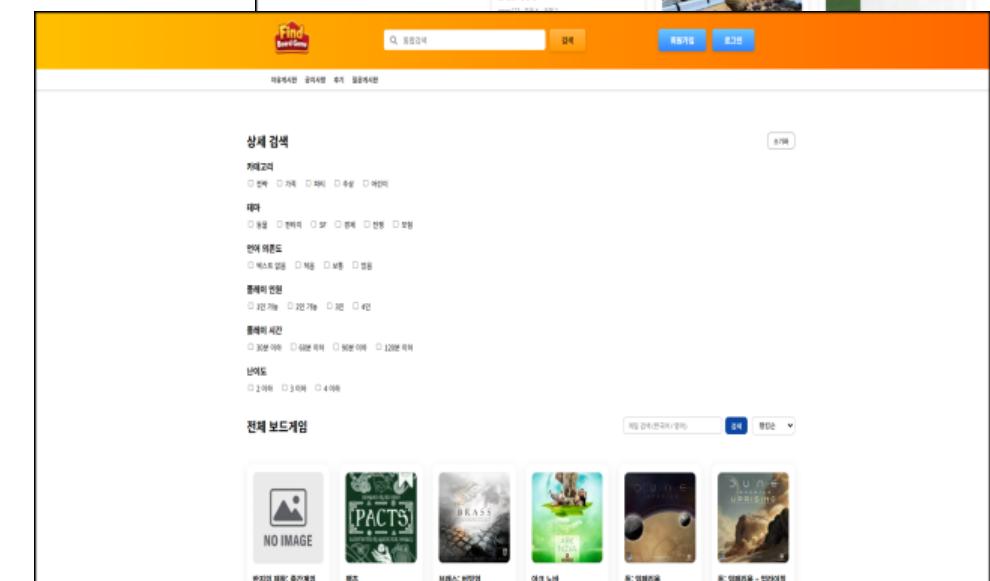
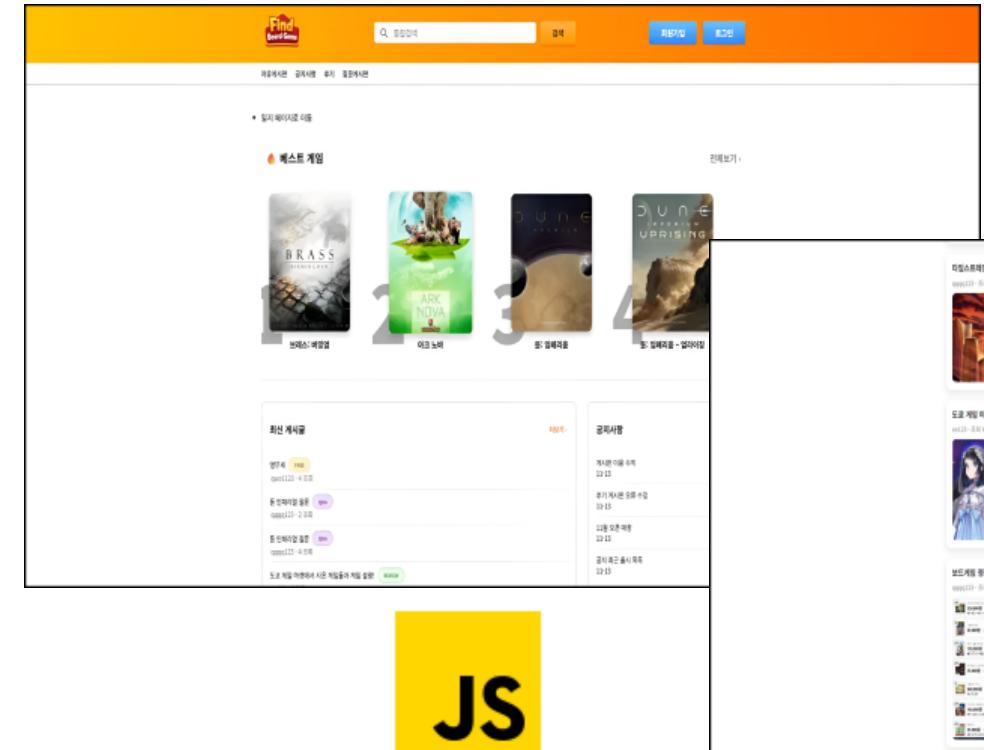
선택 이유

- 스프링 부트만으로 프론트, 백을 구분한 프로젝트를 만들고 싶었습니다.
- 화면 렌더링은 Thymeleaf로 구성하고, 맷글/검색/좋아요/알림처럼 상호작용이 많은 영역은 JavaScript로 부분 비동기 처리했습니다.

구현 포인트

- 페이지 단위로 JS 파일을 분리해 기능 단위로 관리
- fetch 기반 JSON API 연동으로 화면 부분 간신
- 동적 DOM 렌더링 후 이벤트 연결을 위해 전역 바인딩 / 방어 로직 적용
- Spring Security 환경에서 발생하는 401/403 + HTML 응답(리다이렉트) 예외 케이스 대응까지 포함

SSR(Server Side Rendering)의 안정성과 JS의 인터랙션을 결합해 “안정성 있고, 확실히 동작하는” 구조로 구현했습니다.



Technical Choices

프로젝트에 맞게 기술을 선택하고, 역할을 분리 했습니다.

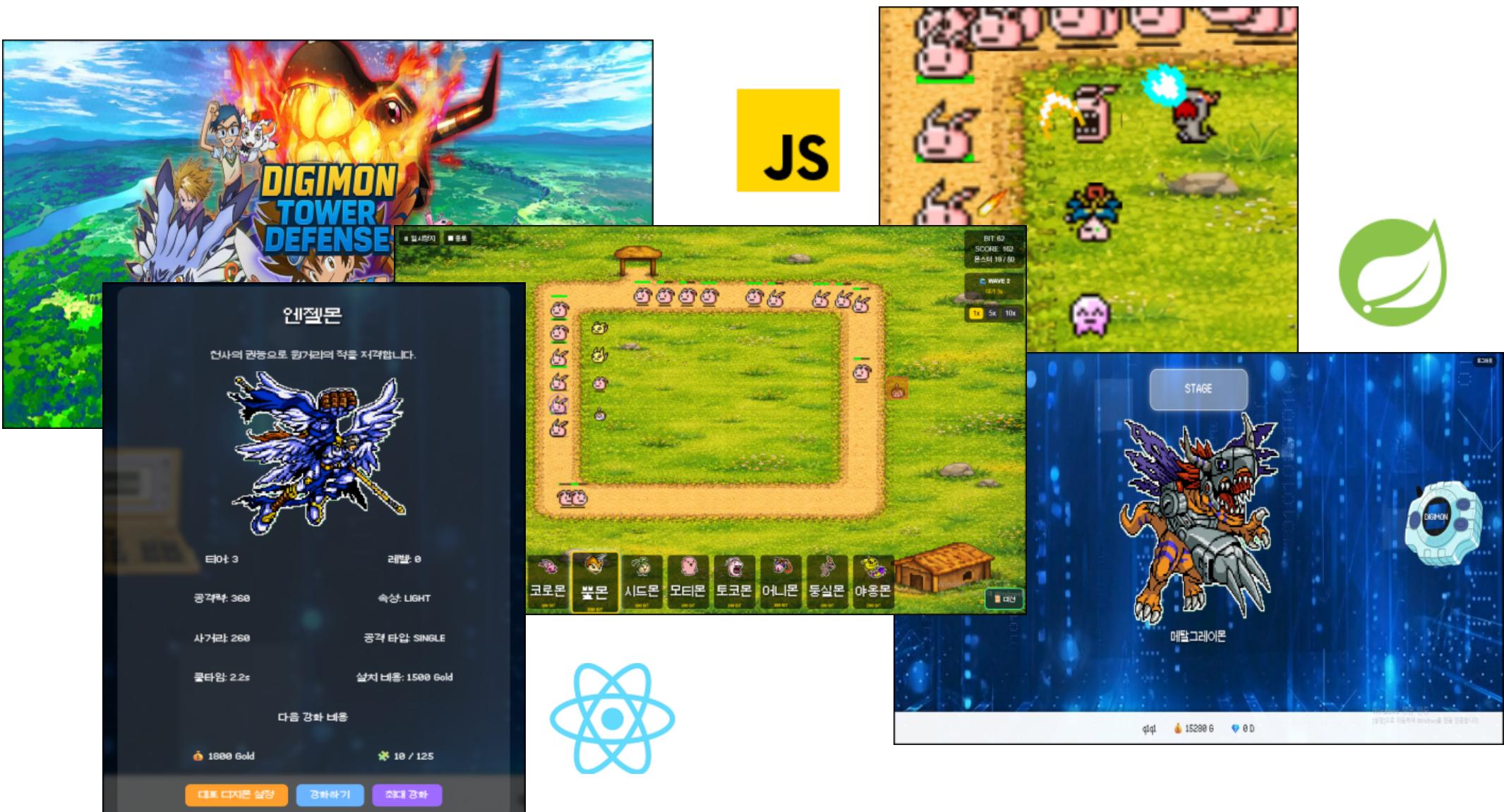
2) DTD – React 중심 (팀 프로젝트 / Frontend 담당)

선택 이유

- 게임은 “페이지 전환”보다 상태 흐름과 UI 업데이트가 핵심이라 SPA(Single Page Application)가 적합했습니다.
- React를 통해 UI를 컴포넌트 단위로 쪼개고, 상태 변화에 따라 화면이 자연스럽게 갱신되도록 구성했습니다.

담당 영역 (프론트 전반)

- 로비/인벤토리/가챠/강화/진화 UI 구현
- Canvas 기반 게임 화면 구성과 UI 연동
- 상태 변화(웨이브/배속/승패/자원)를 화면에 안정적으로 반영



“시간 흐름 기반 상태 변화”를 React 구조에 맞게 설계하여 프론트에서 게임 UX를 완성했습니다.

Final Thoughts

My Approach

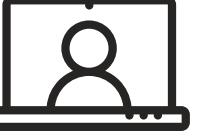


프로젝트를 통해 배운 점

기능 구현보다 구조 설계의 중요성을 경험했습니다.

상태 변화가 많은 화면에서 데이터 흐름을 설계하는 방법을 고민했습니다.

선택한 기술을 단순 사용이 아닌, 이유를 설명할 수 있는 수준까지 이해하려 노력했습니다.

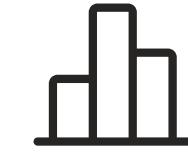


웹 개발을 시작하며

언어에 머무르기보다,
문제를 이해하고 구조로 해결하는 개발자로 성장하고 싶습니다.

새로운 기술을 배우는 것 자체보다,
왜 이런 구조가 필요한지 설명할 수 있는 개발자가 되는 것을 목표로 합니다.

다양한 환경에서도 빠르게 적응하며
팀에 실질적으로 기여할 수 있는 개발자가 되겠습니다.



기술의 확장

새로운 기술과 도구를 열린 자세로 받아들이며,
변화하는 환경 속에서도 빠르게 적응하고자 합니다.

AI와 같은 기술 또한 도구로 이해하며
실질적인 문제 해결에 활용하고 싶습니다.

Thank You

지금까지 읽어주셔서 감사합니다.

GitHub : <https://github.com/wsanghun> / E-mail : zlwm0000@naver.com