

2025.12.02 / Darius

Blind Chicken Market

“가장 늦게, 가장 용감하게. 블라인드 치킨 마켓에서 승리하세요.”

목차

01

프로젝트 개요

주제 선택 이유, 해결방안

02

주요 기능 & 시연

기술스택, UI 흐름, 주요 코드

03

중간발표와 다른점

...

04

트러블슈팅 사례

겪은 문제와 해결법

Blind Chicken Market

“가격 협상과 경쟁이 부족한 기존 중고거래 시장에, 실시간 경매로 공정한 경쟁과 재미를 더하다.”

✓ 가격 경쟁 요소 부족

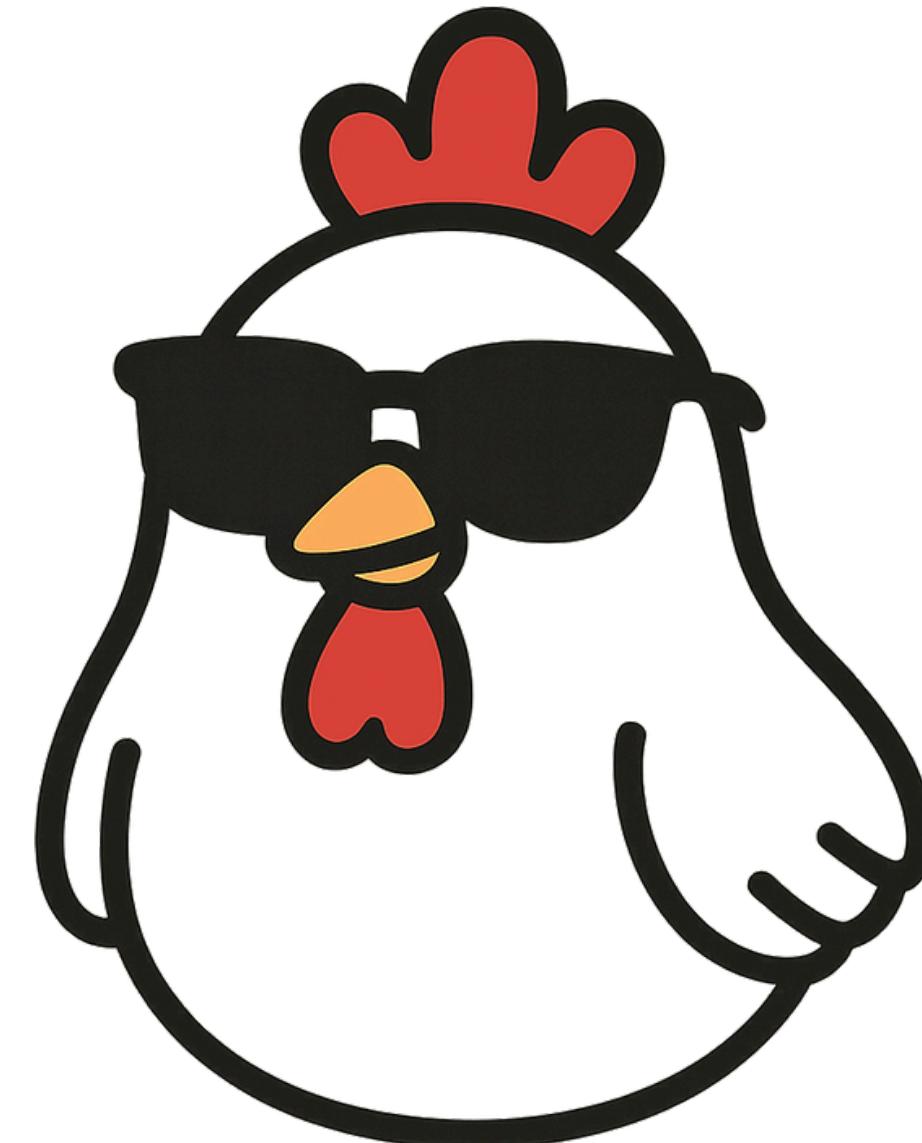
→ 구매·판매자 모두 “적당한 가격”을 찾기 어려워 거래 성사율이 낮아짐.

✓ 거래 경험의 단조로움

→ 실시간성이 없고 흥미 요소가 없어 참여도·몰입도 감소.

✓ 가격 협상 과정에서의 신뢰도, 공정성 문제

→ 판매자·구매자 간 개인 메시지로 협상이 이루어져
가격 조작, 정보 비대칭, 불공정 이슈가 발생하기 쉬움.



익명 기반 실시간 경매 시스템 도입

- ✓ WebSocket 기반 실시간 경매 기능 → 공정한 경쟁 유도
- ✓ 닉네임 기반 거래 → 개인 정보 보호 + 거래 신뢰 강화
- ✓ 게임적 요소 추가 → 거래 경험에 재미 제공



01 프로젝트 개요 - 팀원 소개

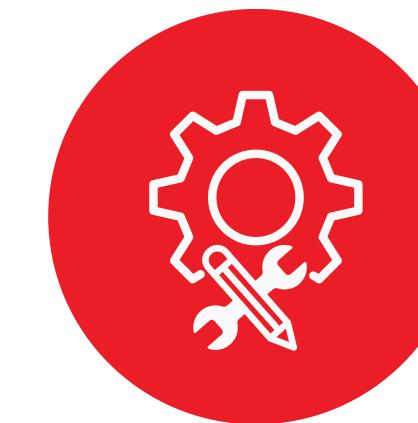
PM, FE | 남경진

- 프론트 구조 설계 및 핵심 컴포넌트 구현
 - 페이지 레이아웃·스타일링 및 프로젝트 관리
-



FE | 최태웅

- 마이페이지, 비밀번호 찾기 기능 구현
-



BE | 이유진

- API·아키텍처 설계 및 WebSocket 실시간 경매 개발
 - 결제·보안 로직 구현과 서버 디버깅
-



DevOps | 정성훈

- AWS 기반 인프라 구축 및 CI/CD 자동화
 - DB 설계·최적화 및 서버 운영
-

Frontend

- Next.js + TypeScript
 - Tailwind CSS, shadcn/ui
 - Axios
-

Backend

- Spring Boot & JPA
 - Real-time: STOMP (WebSocket)
 - Database: PostgreSQL, Redis(NoSQL)
-

Infra/DevOps

- AWS EC2, Route53
- Caddy, CI/CD(Cronjob/Systemd)

02 기능 시연 - UI 흐름 URL

STEP 1

회원가입

플라인드 치킨 마켓에 오신 걸 환영합니다.

이메일	<input type="text" value="your@example.com"/>
비밀번호	<input type="password" value="*****"/>
비밀번호 확인	<input type="password" value="*****"/>
닉네임	<input type="text" value="익명으로 사용할 닉네임"/>
전화번호	<input type="text" value="01012345678"/>
<input type="checkbox"/> 서비스 약관과 개인정보처리방침에 동의합니다	
개정 만들기	
이미 계정이 있으신가요? 로그인	

로그인

플라인드 치킨 마켓에 돌아오셨습니다

이메일	<input type="text" value="your@example.com"/>
비밀번호	<input type="password" value="*****"/>
로그인	
계정이 있으신가요? 회원가입	
비밀번호를 잊으셨나요?	

회원가입 & 로그인

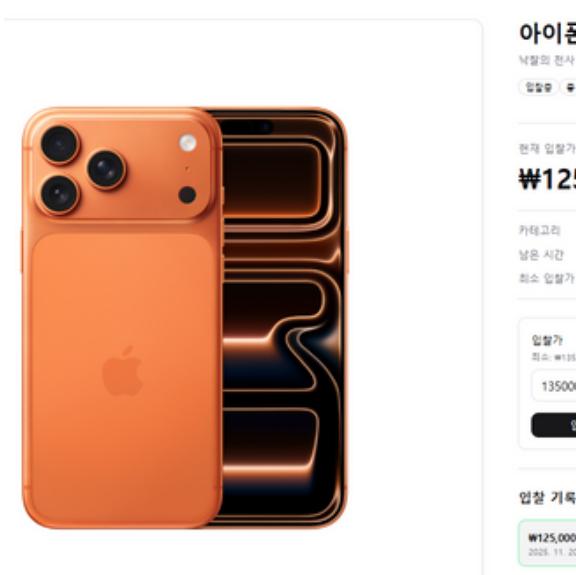
- 닉네임 기반 회원가입
 - 안전하고 빠른 거래 준비 완료

STEP 2

상품 등록 & 리스트 확인

- 판매자가 상품 등록
 - 사용자가 경매 가능 상품 리스트 확인

STEP 3



상품 상세 & 입찰 참여

- 상세페이지에서 상품 정보 확인
 - WebSocket 기반 실시간 입찰 참여
 - 입찰가 실시간 변동 및 경쟁 상황 확인

STEP 4

경매 종료 & 낙찰 확인

- 경매 종료 시 자동 낙찰 처리
 - 낙찰자 확인

01 UI/UX 개선 및 표시 오류 수정

02 회원 관리 및 알림 기능 강화

03 입찰 및 경매 로직 개선

03 중간발표와 다른점 - 결제 페이지 추가

NEW

낙찰은 끝이 아닌 거래의 시작입니다.

✓ 서비스 흐름 완성

→ 낙찰(Bidding)에서 끝나는 것이 아니라,
실질적인 거래 완료까지 이어지는 End-to-End 프로세스 구현.

✓ 직관적인 UI

→ 결제 금액, 배송지 정보를 한눈에 볼 수 있는 직관적인 레이아웃 구성.

✓ 다양한 결제 수단

→ 토스, 신용카드 등 PG사 연동을 통한 사용자 편의성 증대.

The screenshot shows a user interface for a bidding and payment system. At the top, there's a navigation bar with a logo, search bar, and account links. Below it, a large red 'NEW' banner is displayed. A message below the banner reads '낙찰은 끝이 아닌 거래의 시작입니다.' (The bidding is not the end, it's the beginning of the transaction). Three red circular checkmarks highlight features: '서비스 흐름 완성' (Service flow completion), '직관적인 UI' (Intuitive UI), and '다양한 결제 수단' (Various payment methods). The main area contains two forms: '배송 정보' (Delivery information) and '결제 정보' (Payment information). The delivery form includes fields for name, phone number, address, and delivery method. The payment form shows a note about being in a test environment, various payment method buttons (Toss Pay, N Pay, Toss Pay, PAYCO, etc.), and a summary table for the purchase. The summary table details the item as a '프리미엄 티셔츠 인기 상품 판매' (Premium t-shirt popular product sale), with a total amount of ₩15,100.

항목	내용
낙찰가	₩11,000
배송료	₩3,000
수수료	₩1,100
총액	₩15,100

트러블1

로그인 유지 문제

[상황]

- Next.js로 JWT 기반 로그인 구현
- 보안을 고려해 accessToken을 메모리에서만 관리
- 새로고침 시 토큰이 사라져 로그인 유지가 되지 않는 문제 발생

[원인]

- 메모리 저장 방식(RAM)은 페이지가 새로 로드되면 값이 초기화됨
- 재요청 시 API 헤더에 토큰이 포함되지 않아 인증 실패 발생

[해결]

- 로그인 성공 시 accessToken을 localStorage에 저장
- 새로고침 후에도 토큰을 복원하여 지속적인 로그인 상태 유지

트러블2

서버 재가동 시 스케줄링 초기화

[상황]

- 서버 재시작 시 인메모리 타이머 초기화
- 경매 종료 이벤트가 누락되고 DB 폴링(Polling)으로 부하 증가

[문제]

- 메모리 기반 스케줄링은 서버 재시작 시 데이터 소실
- DB 폴링으로 네트워크 트래픽 과다 발생

[해결]

- Redis + Redis ZSet 도입
- 서버 재시작 시에도 스케줄 초기화 문제 해결
- DB 네트워크 트래픽 부하 감소 및 안정적 경매 종료 보장

트러블3

URL 직접 입력을 통한 비인가 접근(IDOR) 방지

[상황]

- 결제 페이지는 /payment/101처럼 URL에 주문 ID가 포함됨.
- 내가 낙찰받지 않은 상품(예: 101번)의 URL을 브라우저 주소창에 직접 입력하여 접속 시도

[원인]

- 단순히 프론트엔드에서 버튼을 숨기는 것만으로는 주소창을 통한 강제 진입을 막을 수 없음
- 보안 장치가 없으면, 타인의 결제 정보를 보거나 대신 결제를 시도하는 심각한 보안 취약점 발생

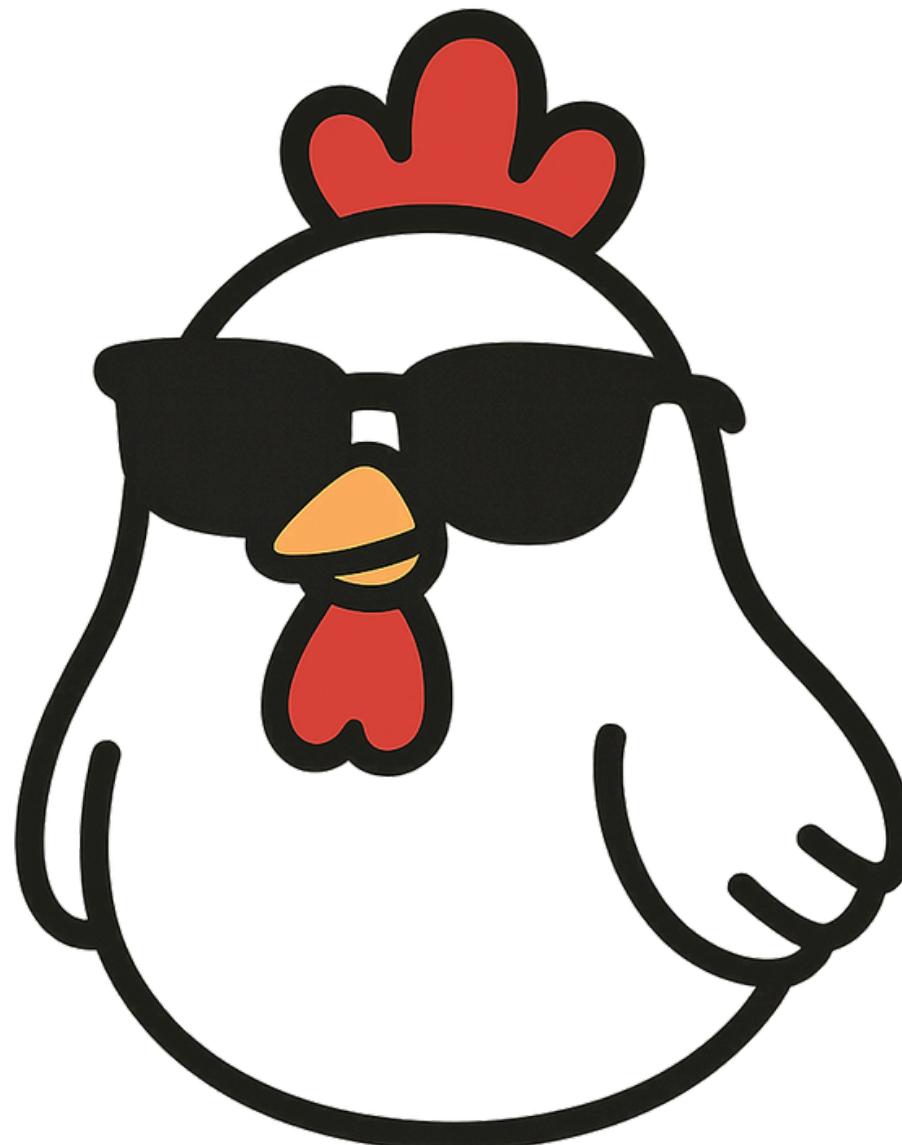
[해결]

API 레벨의 권한 검증 + 프론트엔드 강제 리다이렉트 이중 방어

- 페이지 진입 시 주문 정보 조회 API 호출
- 백엔드에서 403 Forbidden 에러 반환 (본인 주문이 아닐 경우)
- 프론트엔드에서 이 에러를 감지하여 즉시 경고창 띄운 후 메인으로 강제 이동

문서 및 산출물

- 요구사항 정의서
- 시스템 설계도
- UI/UX 설계 시안
- 트러블슈팅 및 개선 기록
-  GitHub Repository (README 포함)



2025.12.02

Q&A

Darius

2025.12.02

감사합니다

Darius