



김대현

프론트엔드 개발자 / 3년차 명지대학교 컴퓨터공학과 학사

Website. <https://frank-town.com>

Email. eogus6242@gmail.com

GitHub. <https://github.com/meohyun2>

‘가장 중요한 개발 역량은 팀 워크’ 라는 원칙으로 성장 중인 개발자, 김대현입니다.

핀테크 스타트업에서 인턴으로 일하며 일정이 밀린 경험이 있었고, 이를 자책하며 개인 회고로 작성해 공유했습니다. CTO님은 ”당연히 실수할 수 있죠! 다만, 팀의 관점에서 해결 방안을 고민해 보세요.”라는 피드백을 주셨습니다.

이 경험은 **‘해결 방안이란 팀을 위한 고민에서 나와야 한다’**는 중요한 깨달음을 주었고, 이후 제 협업의 기준이 되었습니다.

이후 해당 회사에 입사해 디자인 시스템을 구축하고, Web3로 피뎀한 이후 온체인 선물 거래소와 DNS Vault 등 DeFi 서비스 런칭에 기여했습니다. 프론트엔드 외에 도 스마트 컨트랙트 개발과 백테스팅에 참여해 팀에 필요한 기능을 서비스로 배포했고, ‘온체인 선물 거래소’의 UX를 분석해 가이드 문서를 작성하고 세션을 진행해 팀원들의 이해를 도왔습니다.

이후 학업을 마친 후 AI 스타트업의 풀스택 개발자로 일했고, 현재는 Web3 도메인에 집중하며 Rise 체인의 NFT 거래소에서 빌더로 활동하며 개발하고 있습니다.

팀을 위한 엔지니어링과 성장, 그것이 제가 개발자가 된 이유입니다.

Careers

커리어노트

Fullstack Engineer 2024.11 ~ 2025.04

LLM을 활용한 포트폴리오 제작 서비스를 유지보수 하였습니다.

| 포트폴리오 제작 웹 서비스 리팩토링 (Next.js)

Challenges

- 외주 개발사에서 받은 MVP 코드를 넘겨받아 개발했고, 프로젝트 구조가 잡히지 않은 상태였습니다.
- 사용자가 LLM을 활용한 포트폴리오 제작 시, Promise 응답을 단순히 대기하는 경험은 불편했습니다.
- MVP 프로덕트에 비해 낮은 FCP(평균 > 2.0s)가 측정되어 성능적으로 개선이 필요했습니다.

Solutions

- Tailwind CSS 코드가 Arbitrary Value로 정의되어 있었기에, 이를 시스템화 하여 Theme Variable로 관리하여 유지보수성을 높였습니다.
- 중복된 로직과 UI를 가진 React 컴포넌트는 분리한 뒤, 레거시 코드를 제거하였습니다. 결과적으로 번들 사이즈를 10% 줄였습니다.
- 포트폴리오 제작 AI 응답을 Stream으로 변경 후, 이를 UI 컴포넌트와 커스텀 훅으로 분리하여 Headless하게 사용할 수 있도록 수정하였고, 유저들의 포트폴리오 제작 이탈율이 20% 감소하였습니다.
- 시드니 리전에 배포 된 Next.js EC2 인스턴스를 제거하고, Vercel로 마이그레이션 하였습니다. 결과적으로, FCP를 0.6초로 단축시켰고, CI/CD를 적용하여 관리포인트를 줄였습니다.

Nest.js 기반 API 서버 리팩토링

Challenges

- Entity의 변경에 PostgreSQL가 자동으로 업데이트 되는 문제가 있었습니다.
- RDS Connection Pool 수가 과도하게 높아지는 이슈가 있어 서비스 장애가 발생했습니다.

Solutions

- typeorm migration 기능을 활용하여, Auto Update 기능을 끄고, Migration 이력을 별도로 관리하도록 변경하였습니다.
- RDS의 커넥션 풀 연결 대기시간을 1분으로 설정하고, Cloudwatch Event와 Lambda를 활용하여 Slack 알람 웹훅을 개발하여 대응하였습니다.

이력서 PDF OCR 기반 포트폴리오 추출 기능 개발

- OpenAPI Assistants API로 PDF 파일을 읽고, LLM 기반 포트폴리오를 제작하여 PPTX로 추출하는 Langchain 기반 파이프라인을 개발하였습니다.

사용 기술: Next.js, TailwindCSS, Recoil, Nest.js, AWS EC2/RDS/Cloudwatch/Lambda Github Actions, LangChain, OpenAI Assistants

주식회사 티클

Frontend Engineer 2020.11 ~ 2023.06

(2021.07 ~ 2022.02 동안, 창업 준비로 인한 사유로 9개월간 휴직기간이 있습니다.)

정기 저축, 해외주식 투자 앱 "티클"과, 다양한 Web3 서비스를 개발했습니다.

| Vault Defi 서비스 개발 (CRA)

Challenges

- 다양한 화면에서 블록체인과 통신하며 정보를 불러오고, 트랜잭션을 발생하는 JSON RPC Provider 관리가 복잡하게 얹혀 있었습니다.
- 또한 Provider가 분산되어 있어, 응답 받은 정보가 불필요하게 업데이트 되는 경우가 많았습니다.
- DeFi 서비스 특성 상 그래프를 나타내는 컴포넌트가 많았고, UI적으로 비슷한 구성을 띄었지만 각기 다른 데이터를 갖고 있었습니다.
- 웹 상 브라우저, 코인 베이스, Unstoppable, WalletLink 등의 다양한 지갑들을 연결이 필요했습니다.

Solutions

- Contract ABI 로딩과 RPC 호출은 Factory Class로, 실질적인 서비스 로직 처리는 Provider Class로 관리하여, 코드의 관리 복잡도를 낮추었습니다.
- Zustand Slice 패턴을 사용하여, 서비스에서 호출하는 13개의 Smart Contract 와 관련된 Global State의 업데이트를 관리하였습니다.
- 차트와 관련된 디자인 요소들을 Composition Pattern으로 개발하고, 서비스에서 활용하는 8개의 차트 컴포넌트에 반영함으로써, 데이터와 디자인 컴포넌트를 Headless 하게 관리하여 코드 결합도를 낮췄습니다.
- React Context API 를 활용하여, 다양한 지갑에 대한 정보와 액션을 처리할 수 있는 Provider를 개발하였습니다.

사용 기술: Create React Web, React.js, Zustand, web3-react, emotion, ethers.js, Apollo Client, React Router

DeFi 서비스 스마트 컨트랙트 개발 및 백 테스트

Challenges

- 신규 DeFi 기능에 대한 이해가 높았기 때문에, 스마트 컨트랙트 개발 업무에 참여하게 되었습니다.
- 투자 전략에 대한 개발 뿐만 아니라, 백 테스트와 시각화가 필요한 상황이었습니.

Solutions

- Solidity 기반 Smart Contract를 개발하였습니다. Hardhat을 기반으로 해당 Smart Contract에 대한 Unit Test, Deployment 코드를 작성했습니다. Arbitrum Mainnet에 배포하여, 전략에 대한 백테스팅을 수행했고, 런칭을 위한 수익률을 검증했습니다.
- Sushiswap의 거래 데이터를 기반으로, 정해진 기간에 전략을 시뮬레이션하여 influxDB에 인덱싱 하였고, 이를 Grafana로 시각화 하는 코드를 개발하였습니다. 시뮬레이션 자료는 마케팅 문서로 포스팅 되었습니다. (<https://medium.com/@neutralfinance/cpmm-delta-neutra-strategy-nusdc-vault-67a2f2c49116>)

사용 기술: Solidity, Hardhat, Typescript, Grafana, InfluxDB

| 저축&해외주식 투자 앱 개발 (React Native)

Challenges

- 사내 Redux 코드가 모두 JavaScript로 코딩되어 있어, 개발 단계에서 에러를 예측하기 어려웠고, Action, Reducer 코드가 관리된지 오래되어, 다른 개발자가 새롭게 Redux를 활용하기에는 보기 어려운 코드가 많았습니다.
- 해외 주식 기능 검색을 담당 했는데, 유저들의 사용률이 높아 최대한 빠르게 검색 데이터를 제공해야 했습니다.
- 포트폴리오 주식 투자 기능 구현 시 신한금융투자 API를 활용하여 유저들의 포트폴리오 투자가 가능하게 설계해야 했습니다.

Solutions

- Recoil을 도입해서, Atom만 따로 정의하고, Recoil API로만 전역 상태를 제어하도록 변경하였습니다. 결과적으로, Redux 러닝 커브를 줄여 개발 생산성을 높였습니다.
- 주식 검색 기능을 서버와의 통신, 연속된 리스트 렌더링, Image 렌더링 세 가지 부분으로 성능 최적화를 진행하여 FCP를 30% 낮추고, 검색어 입력 시 LCP도 20% 개선하였습니다.
 - 검색 결과를 Prefetching 하여, ApolloClient InMemoryCache에 미리 저장하였습니다.
 - FlatList 사용 시 ‘maxToRenderPerBatch’ 옵션을 기기 별 사이즈에 맞게 설정하였습니다.
 - ‘react-native-fast-image’ 라이브러리를 활용하여, 이미지를 캐싱하여 리스트 별 종목 이미지 렌더 속도를 평균 20% 낮추었습니다.
- 신한 금융 투자에서 제공하는 주식 투자 페이지를 ‘react-native-webview’를 통해 앱 내에 임베드하여 설계 하였습니다. 또한, 신한금융투자와 협업하며 API 명세 상 이벤트에 따라 웹뷰와 앱이 구매 체결 완료 이벤트를 통신할 수 있는 이벤트 브릿지를 설계하여 모듈화 하였습니다.

React Native 서비스 디자인 시스템 개발

Challenges

- 기존 디자인 시스템이 존재했으나, JavaScript로 코딩 되었고, 오래 관리되어 리뉴얼 할 필요가 있었습니다.
- 기존에는 배포 된 앱을 통해 디자이너가 디자인을 확인했지만, 시스템 별도로 테스트 할 수 있는 도구가 필요했습니다.
- 디자인 시스템 도입을 위한 명확한 기준이 필요했습니다.

Solutions

- Atomic Design을 참고하여, 가장 재사용성이 높고 쪼개지기 어려운 Atom, Mocule 컴포넌트를 도입 기준으로 삼았습니다.
- 이를 토대로, 디자이너와 새롭게 디자인 시스템을 기획, 개발하였습니다.
- Storybook을 도입하여, 디자인 시스템을 모두가 테스트 할 수 있게 제공했습니다.
- 디자인 시스템을 개발에 쉽게 적용할 수 있도록, 가이드 문서를 진행하여 제공하여 팀 내 생산성을 높이는 데 기여했습니다.

사용 기술: React Native, Recoil, Apollo Client, Redux, Storybook, React Navigation

| 온체인 선물 거래소 웹서비스 개발

Challenges

- 최초 접속 시 첫 로딩이 너무 긴 문제가 있었습니다. 웹 페이지 로딩까지 평균 2초가 소요되었습니다.
- 2개월이라는 짧은 시간 내, 온체인 선물 거래소라는 생소한 도메인을 이해하여 개발해야 했고, 특히 디자이너 분들이 이해에 어려움을 겪었습니다.

Solutions

- Webpack-bundle-analyzer를 통해, 프로젝트의 번들 구조를 분석했고 총 두 가지 방법을 통해 결과적으로, 번들 크기를 10% 줄여 최적화 하였습니다.
 - RPC 호출 라이브러리를 Web3.js와 Ethers.js 모두 사용했는데, 번들 사이즈가 높은 Web3.js를 제거하고 Ethers.js로 통합하였습니다.
 - React Router 활용한 페이지 중 초기 빠른 로딩이 필요없는 페이지를 Lazy Loading 후, Webpack에서 스플릿 청크로 나누어 사이즈를 줄였습니다.
- 벤치마킹하는 온체인 선물 거래소의 모든 UX를 정리하고, 발표자료로 만들어 사내 세션을 열어 공유하여 프로젝트 진행에 기여했습니다.

사용 기술: React, swr, apollo-client, emotion, ethers.js, react-router

Projects

마이폴리오 2023.06 ~ 2023.08

학생부 전형 준비 AI 서비스 '마이폴리오'의 MVP 서비스를 풀스택 개발하였습니다.

Challenges

- 프론트엔드에서 API DTO를 일일이 반영하는 작업은 휴먼 에러가 발생할 수 있고, 번거로운 작업이었습니다.
- 비 개발자로만 구성된 팀에서 팀 내 개발을 모두 혼자 진행하였고, 각 팀원들끼리 협업하고, 코딩할 수 있는 기준이 필요했습니다.

Solutions

- 개발 시 ‘swagger-typescript-api’ 를 활용하여, DEV 서버에서 제공하는 swagger-json를 받아와 typescript 인터페이스를 codegen 하여 사용하였고, 편리함과 휴먼 에러율을 감소시켰습니다.
- 가장 기본적인 eslint, prettier 세팅부터, Typescript Style 가이드 및 배포 단계 문서화 까지 진행하였습니다. 주 1회 코딩에 어려움을 느끼는 AI 엔지니어 분들과 함께 페어 코딩을 하며, 팀원들의 문제를 함께 해결했습니다.

사용 기술: React, Vite, Nest.js, Serverless Framework, DynamoDB, PostgreSQL

와디 2024.03 ~ 2024.12

크로스핏 기록 측정을 위한 앱을 기획, 개발하였습니다.

Challenges

- 유저 입장의 UX를 고민하며, 기록 측정 시 육체적으로 힘든 유저들의 입장에서, Swipe, Touch 기반의 기록 측정 컴포넌트를 디자인 하였습니다.
- 디자이너와 동료 개발자들의 버전 별 테스트 환경이 필요하였고, 이에 대한 관리 시스템이 필요했습니다.

Solutions

- 운동 시간과 강도를 측정하는 컴포넌트를 모두 Swiper 형태로 자체 개발하였고, Swipe Action에 따른 Haptic 진동이 울리는 컴포넌트를 개발하였습니다.
- Expo EAS Build를 활용하여 테스트 환경과 프로덕션 환경 모두 클라우드 상에서 관리하였습니다.

사용 기술: React Native, Expo, Nativewind, Recoil, Expo EAS

Awards

오픈소스 컨트리뷰톤 장려상

과학기술정보통신부 2020.11.30

오픈소스 기여 해커톤인 '오픈소스 컨트리뷰톤' 에 참가하여 React Native UI 라이브러리인 “Dooboo-ui” 팀에 참가하였습니다. ChipButton 이라는 컴포넌트를 기여했고, 해당 컴포넌트를 개발하면서 Jest, Testing-Library 기반 Unit Test와 Snapshot Test 코드를 개발했습니다.

Activities

한이음 공모전 수료

정보통신산업진흥원 2020.08

[20_HF300] 클라우드 네이티브 콘텐츠 몰 개발 프로젝트에 참여하였습니다. 마이크로 서비스로 운용되는 전체 아키텍처에서 React 기반의 쇼핑몰 웹 페이지 파트를 맡아 개발했습니다.