서지훈

Software Engineer w/ 4 years of experience

hoonie416@gmail.com 010-3047-2972 linkedin.com/in/jihoon416 github.com/jihoon416

ABOUT

안녕하세요. 동료들과 하나의 공통된 목표를 향해 치열하게 협업하는 과정에서 큰 즐거움을 느끼는 개발자 서지훈입니다.

사용자에게 분명한 가치를 제공하고 좋은 사용성을 갖춰, 사람들이 지속적으로 만족하며 사용할 수 있는 제품을 지향합니다.
예전에는 로딩 상태나 플로우를 하나하나 챙기며 개인의 노력으로 사용성을 높이려 했다면, 이제는 세세한 주의를 기울이지 않아도 좋은 사용성이 기본적으로 나오도록 만드는 구조와 시스템이 더 중요하다고 생각합니다.
앞으로의 일에서는 이러한 시스템을 실제로 만들어내는 것에 대해 더 많이 고민해보려 합니다.

일에서 제가 가장 우선순위로 두는 가치는 함께 일하는 사람들과 즐겁게 일하는 것입니다.
동료들이 서로를 믿고 몰입해서 일할 수 있는 환경을 만들고 싶습니다.
건강하고 즐거운 팀 문화를 잘 만들어가는 주변 분들로부터 배우며, 그런 능력을 기르기 위해 노력하고 있습니다.

WORK EXPERIENCE

프론트엔드 엔지니어 (파트타임)
베스펙스

2025년 9월 - 2025년 12월

- 실험용 피처들이 들어가있는 React SPA 웹뷰 내 피처들 개발
- Supabase DB와 Edge Functions를 통해 백엔드 구현

Frontend Engineer
Halfmore

2025년 7월 - 2025년 9월 (1.5개월)

Expo # Next.js # 반응형 디자인 # 데이터 분석

- Expo 기반 모바일 앱과 Next.js 기반 웹에서 피처를 주로 개발. Express 기반 백엔드 서버도 일부 담당하였음.
- 웹 서비스에 Sentry 에러 모니터링을 구축하고 소스맵을 연동하여, 프로덕션 환경에서도 에러를 추적할 수 있도록 개선함.
- 주로 한글 UI를 개발하다가 처음으로 영문 UI를 개발하면서, 좁은 너비 디바이스에서는 텍스트가 잘릴 수도 있는 이슈를 경험함. 이를 해결하기 위해 flex shrink 값 지정, 너비 기반 동적 폰트 사이즈 조정을 적용하여 좁은 화면에서도 깨지지 않는 시각 경험을 제공함.
- 리퍼럴 플로우를 개선하여, 앱에서 리퍼럴 코드를 포함한 웹 링크를 생성하고 다른 사용자가 해당 링크를 열어 전화번호를 입력하면, 모바일 앱 회원가입 시 리퍼럴 코드가 자동으로 적용되도록 구현함.

- PostHog로 웹 랜딩 페이지 A/B 테스트를 진행하고, 사용자의 스크롤 위치에 따른 이탈률을 분석하여 디자인 개선 지점을 도출함.
- A/B 테스트와 더불어 Branch에도 custom properties로 실험군 정보를 기록하여, 실제 앱 설치율과 앱 오픈율까지 분석함.

프론트엔드 엔지니어
라포랩스

2021년 1월 - 2024년 12월

프로모션 쿠폰 게이지

2023년

React Native # 개발 속도 # 팀워크

- 블랙프라이데이 전사행사 직전 행사 기간 중 구매전환율을 끌어올릴 수 있는 빠른 방법이 없을지 스쿼드에서 고민하였음
- 프론트엔드 헤비한 작업이었는데, 한 주 동안 디자인, 개발과 QA까지 발빠르게 진행하여 개발 완료
- 행사 기간 동안 AB테스트를 진행하여 ARPU 증가를 확인 - ARPU 2.1% 상승
- 미구매 유저에 대한 성과에 아쉬운 부분이 있어, 후속 개선 작업을 통해 ARPU 2.2% 추가 상승
- 작업들로 ARPU 총 4.6% 상승

PDP 체감성능 개선

2023년

React Native # 비즈니스 임팩트 # 성능 개선

- PDP(상품 상세 페이지)를 진입할 때마다 모든 데이터를 새롭게 받아오면서 하얀 화면이 일정 시간 동안 보이게 되는 문제가 있었음
- PLP(상품 리스트 페이지)에서 캐시된 데이터로 페이지 일부를 즉시 보여주면 사용자 체감 성능과 탐색 경험이 개선되어 더 많은 탐색으로 이어질 것이라는 가설을 세움
- 스쿼드 작업들을 하면서 병렬적으로 3개월 동안 꾸준히 개선 작업을 진행함. React Query의 placeholderData, RN의 Image.queryCache, InteractionManager.runAfterInteractions 등을 활용함
- AB테스트 결과, 유의미한 구매전환율 상승 확인 - 구매 전환율 1.86% 상승
- 성능 개선을 통해 유의미한 비즈니스 임팩트를 낼 수 있는 사례를 회사 내에서 최초로 제시하였고, 추후 성능 개선 작업이 필요할 때 설득하는 데 도움이 됨
- 작업 과정에서 코드가 복잡해졌는데, 일부 영역에서는 트레이드오프를 고려하여 성능을 조정하고 코드 품질을 더 중시할 수 있었을 것이라는 아쉬움이 남음

사진 리뷰 모아보기

2024년

Next.js # 웹뷰

- 퀸잇 앱 PDP에서 사진 리뷰를 모아볼 수 있는 화면을 만들어 사용자의 구매 경험을 개선함
- 웹뷰로 만든 화면이지만 사용자가 자주 사용할 화면임을 고려해 웹 플랫폼의 성능을 최적화하여 네이티브와 유사한 경험을 제공하는 트랜지션 효과를 구현함
- 시연 영상

기획전 푸시 수신 동의 컴포넌트 플로우 리팩토링

2023년

Next.js # 비동기 처리

- 야간 수신 동의 플로우를 추가하면서 기존 푸시 수신 동의 플로우를 리팩토링함
- Promise를 await하는 곳과 resolve하는 곳을 분리할 수 있는 createDeferred 유틸리티를 추가함(ES2025의 Promise.withResolvers()에 해당하는 기능). 이를 통해 수신 동의 플로우 단계를 async 함수로 표현하여 여러 UI가 등장하는 플로우를 명령형 비동기 함수로 구현, 가독성을 개선함
- 기존에는 전체 플로우를 한눈에 파악하기 어려웠으나, 이 유틸리티 활용으로 전체 플로우를 한 번에 볼 수 있게 됨

출석체크 비동기 플로우 구현

2024년

Next.js # 비동기 처리

- 출석체크 완료 후 모달을 통해 앱러빈 광고, 쿠팡 방문하기 두 가지 미션 참여를 연속해서 유도하는 기능을 개발함
- 미션 참여는 버튼을 통해 할 수 있으며, 남은 미션이 있다면 연속으로 실행되도록 하는 스펙을 구현해야 했음
- 미션 수행 플로우를 비동기 함수로 정의하고, 플로우 완료 시 리턴값이 다음 수행할 미션으로 resolve되도록 구성함. 이를 통해 추가 미션이 있으면 재귀적으로 함수를 실행하고, 없으면 종료하는 방식으로 개발함
- 중간 임시 상태를 최소화하고 비동기 함수의 결과값으로 표현하여, 코드 파악 시 직관적으로 플로우가 보이도록 개발함
- 스펙을 제거할지도 고민했으나, 미션 참여가 자동으로 유도되는 것이 참여율에 상당한 임팩트가 있을 것이라 판단해 유지. 실제로 미션 참여를 자동으로 유도하지 않는 다른 화면에 비해 참여율이 높게 나타남

Redux에서 React Query로 마이그레이션

2022년

React Native # React Query

- 초기에는 Redux와 Redux Observable 기반으로 코드가 작성되어 있었음(2020~2021)
- 2021년부터 모바일웹에서 새로 작성되는 코드는 일부 React Query를 사용 중이었음
- 2022년에 사내 최대 규모 프론트엔드 프로젝트였던 RN 앱 코드베이스에 React Query를 도입하고, 기존 Redux Observable 패턴을 대체할 수 있는 방법을 제시함
- 시니어 개발자들의 코드리뷰를 받으며 Query 및 Mutation 훅 관리를 위한 코드 컨벤션을 제시함 - 이 컨벤션은 코드베이스 전체에서 사용되다가 2023년에 새로운 컨벤션으로 대체됨

Jotai에 대한 지식 전파

2024년

Jotai # 상태 관리

- Redux에서 관리하던 데이터가 대부분 서버 데이터였기에 React Query로 대체 가능해짐
- 클라이언트 전용 상태만 남게 되어 Redux는 불필요한 보일러플레이트가 많았음. 더 나은 방법이 필요하다는 논의는 있었으나, 클라이언트 상태 관리가 자주 등장하는 문제가 아니어서 우선순위가 낮아 진전이 없었음
- 일부는 Context API를, 일부는 사내 개발된 커스텀 상태 라이브러리를 사용하면서 스쿼드 구성원들이 코드 작업에 어려움을 겪거나 의도치 않은 성능 저하가 발생함

- 여러 상태 라이브러리를 조사한 결과, Jotai가 가장 적합하다고 판단함 (API가 간단해서 동료 엔지니어들의 적응이 쉬울 것 같다는 점 및 다양한 유용한 유틸리티 제공을 큰 장점으로 느낌)
- Jotai를 활용한 상태 관리 방법, 장점, 코드베이스 도입 시 이점 등을 주제로 발표함
- 진행 중인 작업 우선순위로 인해 직접 첫 도입은 하지 못했으나, 동료 엔지니어들이 Jotai의 필요성에 공감하고 도입하여 이후 회사 표준 상태 라이브러리로 자리잡음
- 상태관리 라이브러리에 대한 기술부채 해소의 필요성을 PO에게 설득시켜서 직접 도입까지도 할 수 있었는데 하지 못했던 것은 개선해볼 부분이라고 생각함

SKILLS

장애 및 이슈 대응

다른 동료들이 장애를 대응 중일 때 웬만한 급한 일이 없으면 해결 과정에 같이 참여하였고, 다른 동료의 작업에서 제가 직접 수정할 수 있을만한 이슈를 발견하였을 때 직접 수정하였습니다. 이슈 발생 시 원인을 빠르게 파악하는 능력과 대응하는 능력이 뛰어나다는 피드백을 받았습니다.

- 특정 디바이스에서 상품상세페이지 HTML이 표시되지 않는 현상이 발생했을 때, http로 받아와야 할 파일이 https로 요청되는 문제를 확인함. 서버에서 제공하는 HTML 페이지의 HSTS Policy 변경으로 인한 것을 파악하였으며, 이후 DevOps팀의 인프라 변경 작업에서 비롯된 것임을 확인함
- react-native-netinfo 버그로 인해 네트워크 모드가 잘못 인식되어 React Native 앱에서 React Query가 일시적으로 작동하지 않는 이슈를 수정함
- 크래시를 모니터링하던 중 API 에러로 인해 크래시할 수 있는 RxJS 코드를 발견하고 대응 - 한 달에 2만명 정도 앱 시작 시 죽는 문제가 있었는데 이를 수정함

플랫폼 작업

- React Native 버전 업그레이드 및 파이어베이스 라이브러리, 안드로이드 minSdk 최신화 작업 수행
- CI에서 React Native 빌드 시간 단축을 위해 불필요한 스텝들 제거 및 캐싱 추가
- React Native 프로젝트에서 이미지 애셋들이 들어있는 라이브러리가 tree shaking되지 않고 전체가 번들되면서 불필요한 이미지 애셋들까지 포함되면서 번들 사이즈를 지나치게 키우는 문제가 있었음. tree shaking되는 번들러로의 전환도 고려했으나, 당장 겪는 문제에 비해 비용이 크다고 판단하여 대신 import문을 트랜스폼하는 babel 플러그인 작성

EDUCATION

전기정보공학부 2019년 3월 -

서울대학교

영재학교 졸업 2016년 3월 - 2019년 2월

경기과학고등학교