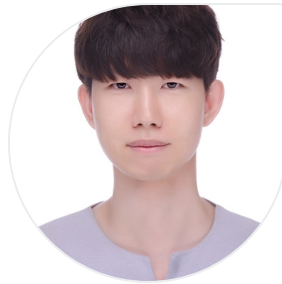




차민재님을 응원해보세요!

이직/구직 중이에요 # 협업 지향 # 목표 지향적 # 직관적 # 커뮤니케이션 # 고객 중심



차민재

신입 개발자

안녕하세요. 개발 커리어를 시작하는 신입 개발자 차민재입니다.
팀원과의 소통을 바탕으로 더 좋은 UX를 가진 서비스를 구현하는 것이 개발에서의 목표입니다.

기술 스택

Java

JavaScript

Spring

Spring Boot

React

MySQL

TypeScript

Python

Django

Vue 3

react-query

zustand

redux-toolkit

msw

경력

주임 • 5센서사업부

아이쓰리시스템

2019.01. ~ 2022.10. (3년 10개월 | 정규직)

추천해요 🍷

제안하기 ✉

댓글 💬

반도체 엔지니어 경력

- 적외선 반도체 양산 공정 관리
- 공정 이슈 / 장비 이슈 대응

프로젝트

상추

기타

2024.02. ~ 2024.04. (3개월)

상추 프로젝트는 농수산물 빅데이터를 활용하여 레시피의 최저가를 제공해주는 서비스입니다.
기술적인 특이점은 다음과 같습니다.

- **FE : 하이브리드 앱을 제작하여 웹뷰 형태로 서비스 제공**
 - 서비스 특성상 웹 접근성보다는 모바일 접근성이 더 좋다고 판단되었습니다.
 - 접근성이 좋을 뿐 모바일 필요성이 존재하진 않기에 웹 개발을 바탕으로 하이브리드 앱으로 구현했습니다.
 - 하이브리드 앱 구현으로 사용자가 앱을 업데이트 하지 않아도 업데이트 된 서비스를 경험할 수 있습니다.
- **BE : 농수산물 빅데이터 분산 처리를 통한 데이터 가공**
 - 빅데이터 분산 처리 개념과 기법을 배웠습니다.(분할 정복 알고리즘)
- **DevOps : MSA 도입**
 - MSA를 도입하며 여러 서버에서의 인가 문제가 발생했고, 이를 해결하기 위해 passport를 이용하여 여러 서버에서의 인가를 진행했습니다.

개인적으로는 아래와 같은 경험을 했습니다.

불편한 UX로 제공되는 농수산물 카테고리의 정리를 위한 팀 회의

- 문제점
 - 백엔드로부터 받아 온 농수산물 카테고리가 16가지로 분류되어 들어왔습니다.(닭고기, 돼지고기, 소고기 등등)
 - 어떤 카테고리들은 데이터가 두, 세 개나 네, 다섯 개 정도로 적어 사용자 입장에서 다소 불편하다고 판단했습니다.
- 해결책

댓글 💬

- 문제점을 공유 받아 팀 회의를 진행했고 성질이 비슷한 카테고리를 묶어서 데이터를 가공하는 것으로 결론을 냈습니다.
- 16가지 데이터를 6가지로 줄여 UX를 개선했습니다.
- 카테고리를 팀 내에서 재정의하여 사용하게 되어 프론트엔드에서도 상태 관리를 통해 관리할 수 있게 되었습니다. 덕분에 여러 컴포넌트에서 손쉽게 카테고리를 사용할 수 있었습니다.

프론트엔드 CI/CD 구축

- 구축된 인프라 환경에서 프론트엔드 코드의 CI/CD 파이프라인을 구축해볼 기회를 받아 직접 시도했습니다.
- 젠킨스를 이용한 파이프라인을 구축했고, 파이프라인 스크립트에 사용되는 groovy 문법에 대해 배울 수 있었습니다.
- 또한, 개발 이후 직접 서버에 배포되기까지의 인프라 과정을 배웠습니다.

React Native를 이용한 웹뷰 앱 제작

- 서비스 특성을 고려해서 하이브리드 앱이 적합하다고 판단하여 모바일 앱 제작을 시도했습니다.
- 웹뷰 방식이 어떻게 동작하는 지 학습했고, Expo를 이용하여 웹뷰로 앱을 제작하는데 성공했습니다.

시그니처

기타

2024.04. ~ 2024.05. (2개월)

시그니처 프로젝트는 삼성 월렛의 결제 기능을 바탕으로 탄생한 대리 결제 서비스입니다.

기술적인 특이점은 다음과 같습니다.

- **FE : 완전한 모바일 앱 제작, 개발 시 MSW를 이용한 API 모킹**
 - 시그니처 앱과 삼성 월렛 앱을 서로 앱 링크로 연결해야 하며 이외에도 다양한 모바일 접근성은 필요하나 웹 필요성은 없는 서비스였기에 모바일 앱으로 구현을 결정했습니다.
 - MSW를 도입하여 API 개발이 되기 전에 모킹된 서버에서 API 통신 작업을 진행했습니다.
- **BE : 결제 시스템의 이해, 결제 로직 구현**
 - 일반적인 카드 결제가 아닌 삼성 월렛에 저장된 카드의 결제 방식에 대한 이해가 필요하여 결제 시스템과 삼성 월렛에 저장된 카드가 어떻게 사용되는지를 이해했습니다.
 - PG사의 구현을 제외한 방식으로 결제 시스템을 구현했습니다.

댓글 0

개인적으로는 아래와 같은

상황에 맞는 상태 업데이트 방법 학습

- 문제점
 - 내 카드의 결제 권한을 친구 A에게 주었을 때 카드의 스위치를 on/off 하며 결제 권한을 제어하는 로직이 있습니다.
 - 결제 권한을 on/off 하는 API 문서는 응답이 "성공 or 실패"로만 주어졌고 저 역시 아무 고민없이 API 문서에 맞춰 로직을 구현했습니다.
 - 결과적으로 스위치 on/off 작업 시 스위치 상태 데이터 전달 -> 백엔드로부터 성공 응답 -> 현재 스위치의 상태 값 조회 요청 -> 스위치의 변경된 상태 응답의 과정을 거쳐 사용자가 스위치 클릭 후 상태 변경까지 대략 1초 가까운 시간이 걸리며 좋지않은 UX를 제공했습니다.
- 해결책
 - 백엔드와 상태 업데이트 통신을 할 때는 크게 세 가지 방식을 선택할 수 있다는 것을 배웠습니다.
 - 첫 번째, 백엔드로부터 업데이트 성공 메시지에 업데이트된 상태 응답으로 받기
 - 두 번째, 제가 선택한 방식으로 상태 업데이트 이후 재조회를 통해 데이터 받기
 - 세 번째, 프론트엔드에서 업데이트한 상태로 상태 직접 변경하기
 - 각각의 방식은 장단점이 서로 존재하는데 대체적으로 첫 번째 방식이 가장 안정적이고 정석적이라 1순위로 고려를 하고 만약 서버의 상태가 좋지 못 할 경우 나머지 방식을 고려해야 함을 배웠습니다.
 - 이번 트러블의 경우에는 사용자에게 더 빠른 피드백을 위해 세 번째 방식으로 고려했어야 한다고 판단했습니다.

API 문서에 대한 파라미터 크로스 체크

- 카드의 결제권한을 on/off 하는 API 문서에 필요한 파라미터가 카드의 id 값만 넘겨주는 것으로 작성되었습니다.
- 제가 생각한 로직으로는 카드의 id 값과 권한 상태를 변경할 사용자의 id 값이 같이 필요하다 판단했고, 이를 해당 백엔드 팀원에게 크로스 체크 요청을 했습니다.
- 백엔드에서 데이터베이스 설계를 할 때 카드마다 권한이 부여된 사용자에게 붙인 id를 카드 id로 명명하였고, 해당 id만으로 결제권한의 on/off가 가능했습니다.
- 혹시라도 발생할 수 있는 문제 상황을 미리 백엔드와 크로스 체크해본 경험이었습니다.

모바일 앱 개발 방식에 대한 이해

- 순수한 모바일 앱으로 개발하면서 웹 개발과 구조적으로 많은 차이가 있음을 배웠습니다.
- 특히, 개발 초기에 라우팅 구조 설계에 대해 많이 헤맸고 모바일의 라우팅은 스택 형태로 설계됨을 학습했습니다

맛동산

기타

2024.01. ~ 2024.02. (2개월)

맛동산 프로젝트는 맛집을 저장, 기록하고 사용자간 공유할 수 있게 해주는 맛집 SNS입니다.

기술적인 특이점은 다음과 같습니다.

- **Git-flow 전략으로 브랜치 관리**
 - Git-flow 전략을 처음으로 도입해본 프로젝트이며, 개발 상태에 맞게 브랜치를 명명하고 팀 컨벤션에 맞춰 진행한 프로젝트입니다.

개인적으로는 아래와 같은 경험을 했습니다.

FE 개발 환경 구축 및 Lint 설정

- FE 리더를 담당하며 npm을 기반으로 React 개발 환경을 구축했습니다.
- FE 코드 컨벤션과 스타일 가이드를 설정하고 각각 ESLint와 Prettier를 이용해 공통된 스타일로 관리하고자 노력했습니다.

포트폴리오

URL

 [포트폴리오](#)

 [깃허브](#)

 [블로그](#)

교육

삼성전자주식회사

2023.07. ~ 2024.06.

졸업

사설 교육 | 삼성 청년 SW

댓글 

멀티캠퍼스

2022.12. ~ 2023.06.

졸업

사설 교육 | [멀티잇] 채용연계 풀스택 개발자 취업캠프(java)

충남대학교

2013.03. ~ 2019.02.

졸업

대학교(학사) | 물리학과

외국어

영어

일상 회화 가능

자기소개

안녕하세요. 개발 커리어를 시작하는 신입 개발자 차민재입니다.

팀원과의 소통을 바탕으로 더 좋은 UX를 가진 서비스를 구현하는 것이 개발에서의 목표입니다. 반도체 공정 엔지니어로 근무하다 건강상의 이유로 퇴사하고 대학 시절 흥미와 좋은 성적이 있던 코딩에 다시 관심을 가졌습니다. 부트캠프와 SSAFY 교육을 이수하며 웹 개발에 대해 배웠고 거대하게 성장한 JavaScript 생태계에서 개발 커리어를 시작하고픈 목표가 생겼습니다.

웹 서비스를 구현할 때는 항상 UX의 고민이 필요하다고 생각합니다. 프론트엔드 개발자라고 UX를 고려하지 않고 구현한 예쁜 UI는 의미가 없고, 백엔드 개발자라고 UX를 고려하지 않은채 개발하기 편한 로직 구현 역시 잘못된 개발 방식이라고 생각합니다. 모두가 서로 다른 의견을 가지고 일을 하지만 공통적인 "어떻게 UX를 더 개선할 것인가?"를 목표로 서로 회의를 하며 합의점을 찾아나가는 것이 개발자들의 의사소통이라고 생각합니다.

이러한 대표적인 경험을 "상추" 프로젝트를 진행하며 얻었습니다. 빅데이터 처리를 바탕으로 진행된 프로젝트였기에 농수산물의 데이터를 외부에서 받아왔고 이를 저희 서비스에서 표현하는 과정에서 너무 많은 카테고리나 오히려 사료가의 떼어따리다는 의견을 모두가 수용하고 자체적으로 카E | 댓글 0 관들었습니다.

또한, 사용자의 상호작용으로 데이터를 업데이트하는 과정에서 데이터 처리방식에 따라 UX가 극명하게 달라질 수 있음을 "시그니처" 프로젝트를 진행하며 경험했습니다. 이 경험을 통해 데이터 업데이트 과정의 3가지 방식에 대해 학습했습니다.

앞으로의 경험에서도 어떻게 사용자들에게 더 좋은 경험을 줄 수 있을지를 고민하며 성장하고 싶습니다.

Powered by Rallit.