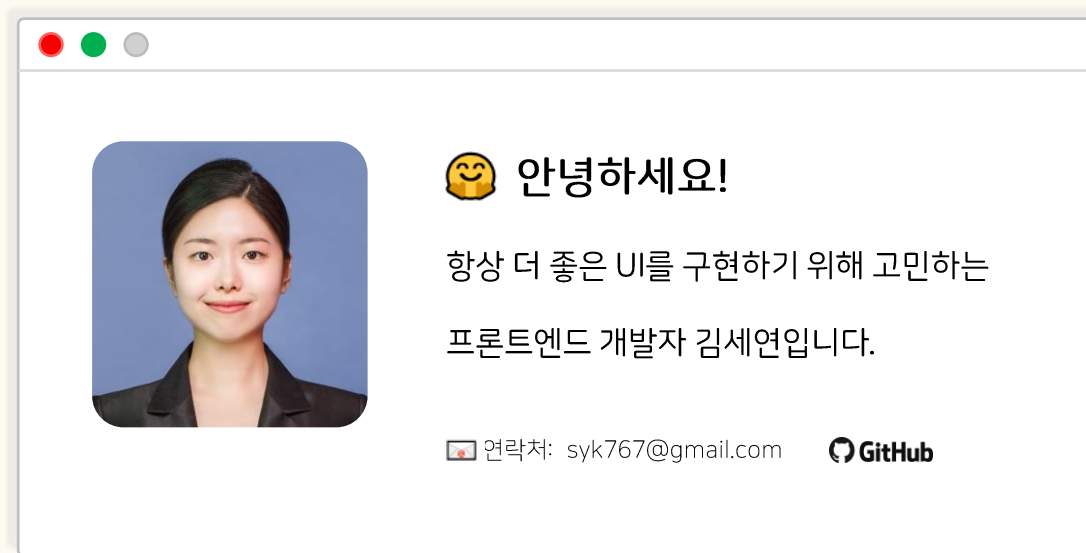


“ 제 몫을 해내는 개발자,
함께 일하는 것이 즐거운 동료가 되고자 합니다.”



경험

1 스파르타 코딩클럽 부트캠프 '항해99' 3기 수료

'99일간, 주 6일 하루 12시간 이상' 공부를 목표로 하는 부트캠프 프로그램입니다.

매주 새로운 팀원들과 공부를 하며 프로젝트를 완성합니다.

React를 활용한 프로젝트 4개를 완성하였습니다.

최종 프로젝트 'Fungap'은 프론트엔드 3명, 백엔드 3명, 디자이너 2명을 팀으로 6주의 기간을 가지고 완성하였습니다.

2 경기도형 대학생 취업브리지, (주)무른모 개발 인턴

(주)무른모는 자동차 부품 제조 기업을 대상으로 MES를 제공하는 기업입니다.

서비스를 이해하기 위해서 자동차 부품 제조 과정의 큰 프로세스에 대해서 학습하였습니다.

고객센터에 올라오는 개선 사항들을 해결하는 업무를 맡았습니다.

업무 프로세스: 이슈 분석 -> 이슈 해결 -> 코멘트 남기고 개발 서버에 적용

- HTML5, CSS3를 활용한 퍼블리싱 작업

- 쿼리문을 수정하여 데이터 오류를 해결하는 작업

👜 경력

- 2021.09 ~ 2021.12
스파르타코딩클럽 부트캠프 '항해99' 3기 수료
- 2020.01 ~ 2020.05
경기도형 대학생 취업브리지, (주)무른모 개발 인턴

🎓 학력

- 2015.03 ~ 2020.08
한신대학교 컴퓨터공학부 (학점: 3.48/4.5)
- 2012.03 ~ 2015.02
인천여자고등학교

🏆 자격증/수상내역


- 정보처리기사 (2019.11)
- 취업희망 기업&직무 분석대회 최우수상 (2019.12)

프로젝트

Fungap : MBTI 관련 콘텐츠 제공 서비스

[바로가기](#)



 기간 | 2021.10 ~ 2021.12 (총 6주)

 주소 | 서비스 주소: [링크](#) / 프론트 깃허브: [링크](#)

프로젝트에서 담당했던 업무

- 관리자 페이지 CRUD 작업.
- 밸런스 게임 페이지 CRUD 작업.
- react-slick 라이브러리를 활용한 슬라이더 구현.
- react-simple-chatbot 라이브러리를 활용한 챗봇 구현.
- Chart.js 라이브러리를 활용한 그래프 구현.
- AWS S3를 활용한 이미지 업로드 구현.
- 프로젝트 중간, 최종 발표 자료 담당.

핵심 경험

- React, Redux를 활용한 웹 어플리케이션 개발 및 CRUD 구현.
- 사용자 피드백을 반영하여 어플리케이션 유지보수 경험.
- 백엔드&프론트엔드 개발자, 디자이너와의 협업 경험.

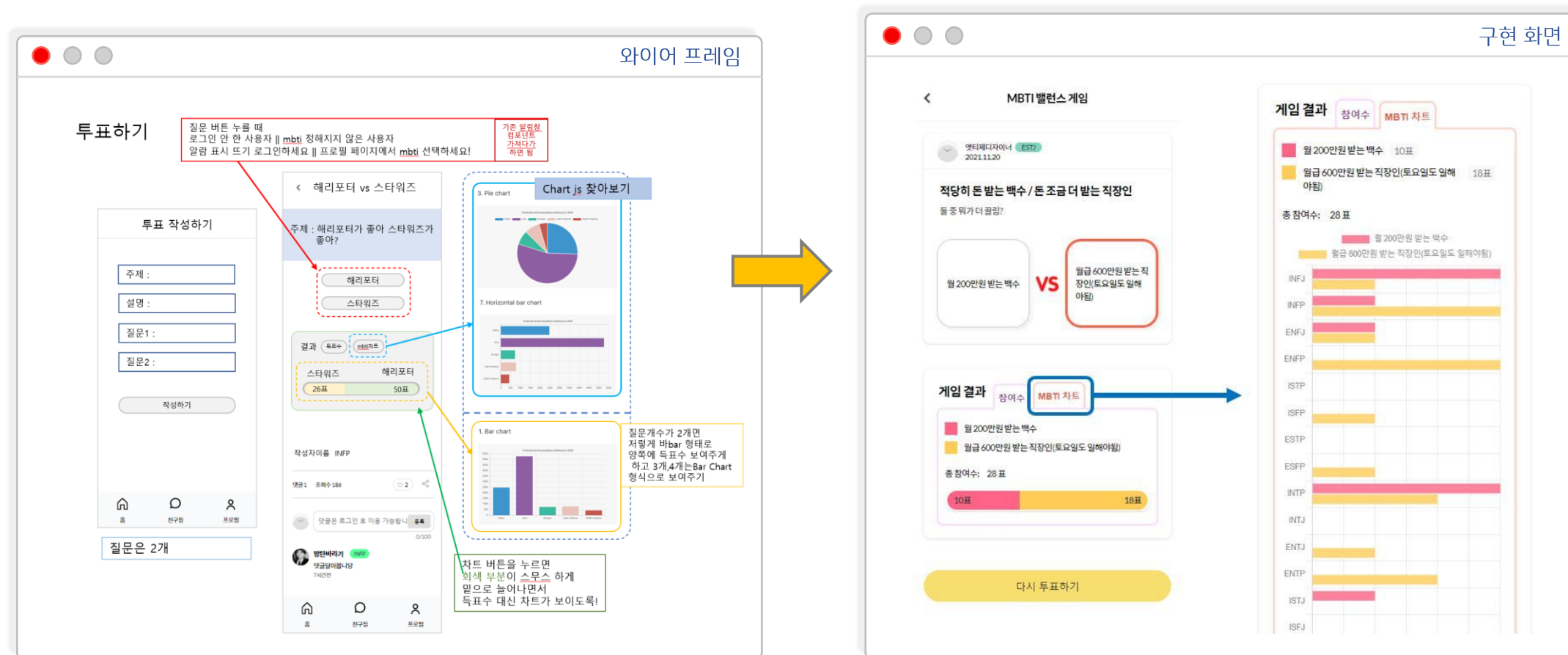


아이디어를 제안하고, 와이어 프레임을 설계하고 그대로 구현할 수 있습니다.

‘항해 99’ 최종 프로젝트에 참여하였을 때, 서비스의 콘텐츠가 부족하다는 생각이 들었고 ‘밸런스 게임’이라는 콘텐츠를 제안하였습니다.

저는 서비스에 필요한 개선사항에 대해서 적극적으로 의견을 낼 수 있으며, UI를 설계하고 설계한 화면을 그대로 구현할 수 있습니다.

이러한 경험을 통해서 앞으로 맡게 될 업무에 대해서도 주체적으로 의견을 내고 좋은 퍼포먼스를 낼 수 있을 것입니다.



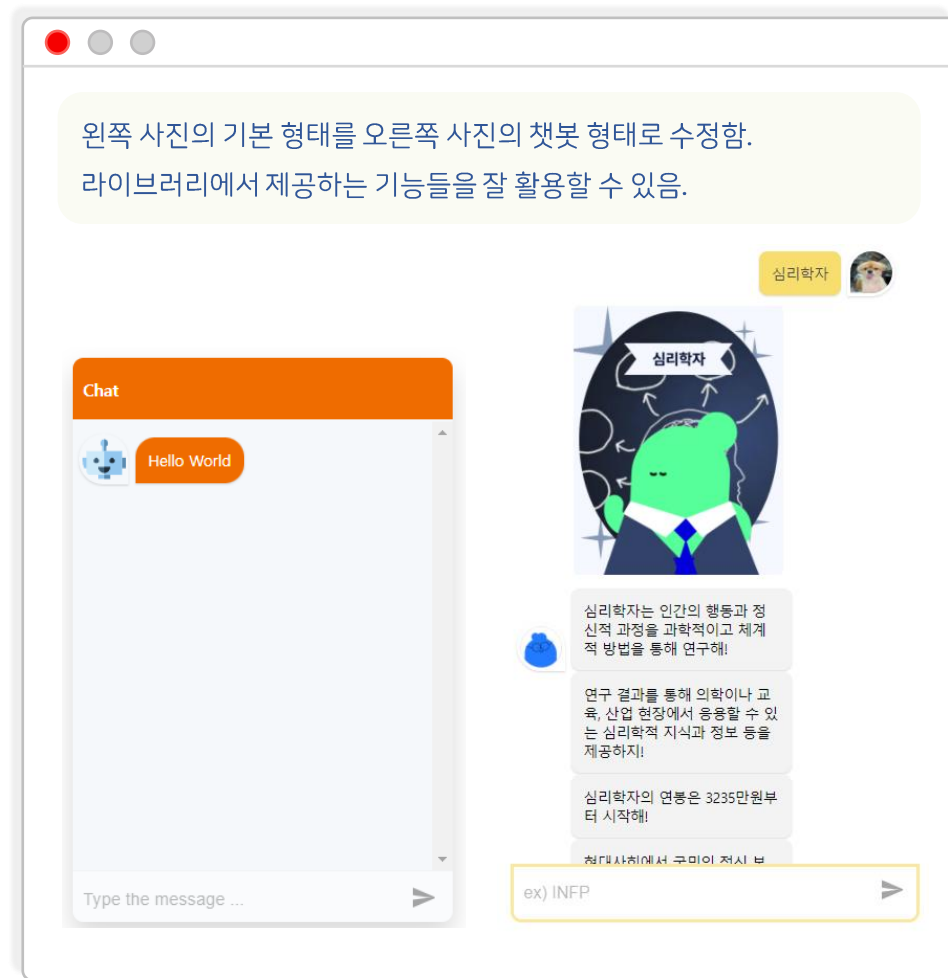
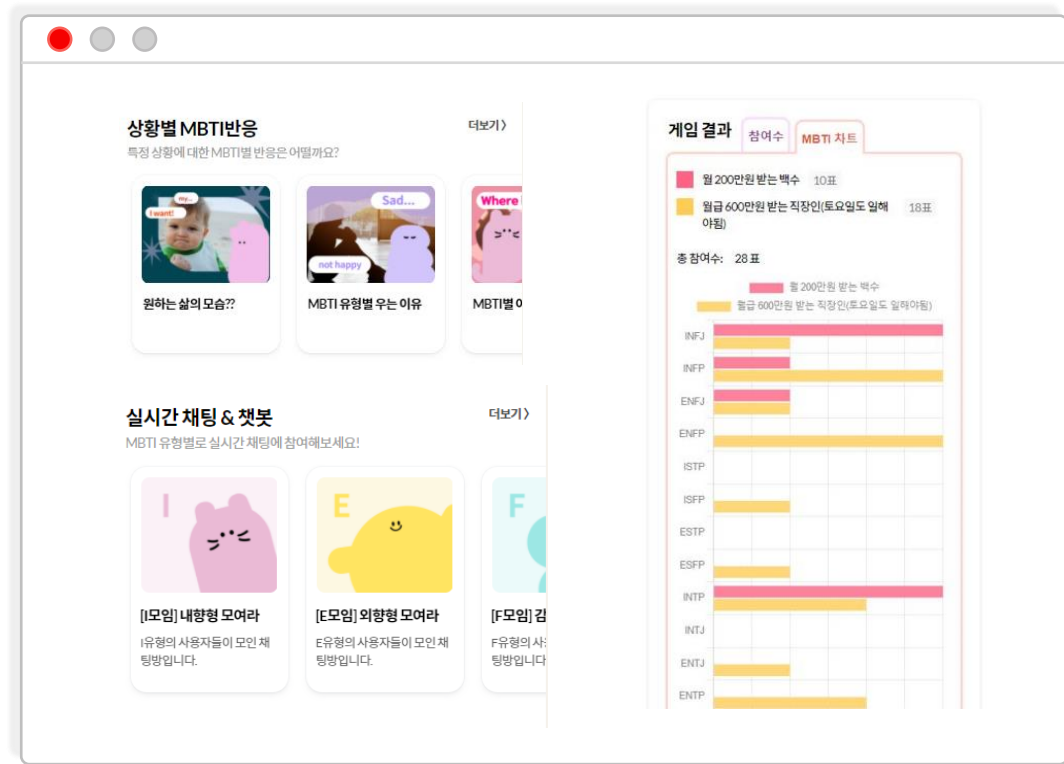


다양한 라이브러리를 활용해서 기능을 구현할 수 있습니다.

'항해 99' 최종 프로젝트에 참여하였을 때, 다양한 React 라이브러리를 사용하여 챗봇, 그래프, 슬라이더를 구현할 수 있었습니다.

이렇게 필요한 기능 구현을 위해 라이브러리를 잘 활용했던 경험이 쌓이면서 라이브러리를 활용하는 것에 거리낌이 없어지고

다양한 기능들을 구현할 수 있는 능력을 높일 수 있을 것입니다.



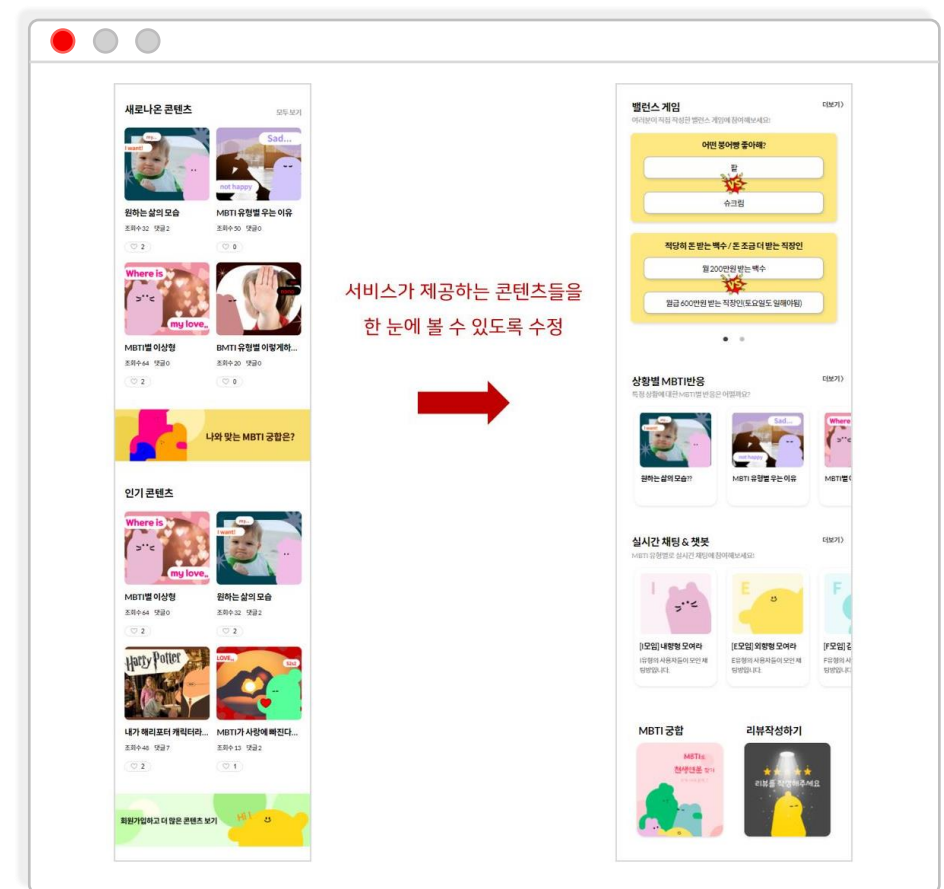
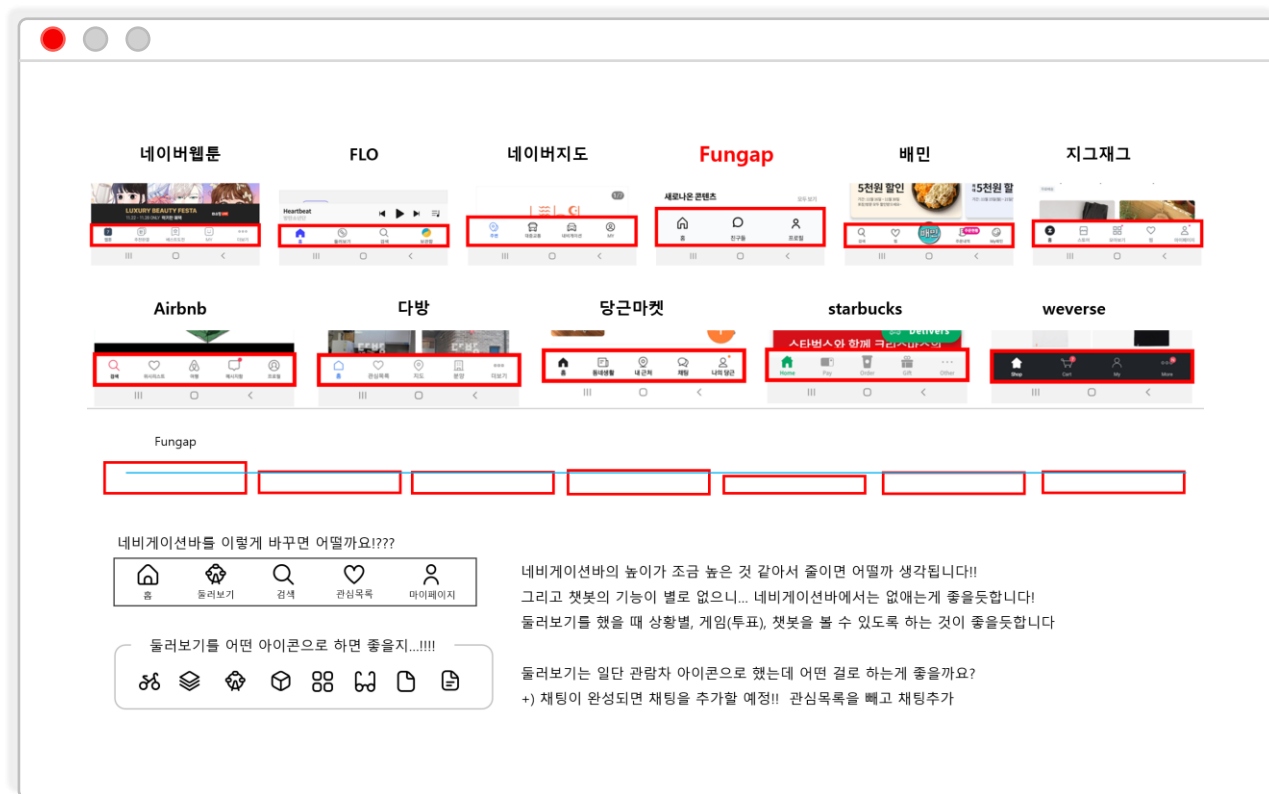
더 좋은 UI 구현을 위해 고민합니다.

‘항해 99’ 최종 프로젝트에 참여하면서, 누군가 시키지 않아도 어떻게 하면 지금의 UI를 더 개선할 수 있을지 고민하였습니다.

기본 네비게이션 바를 실제 서비스들의 네비게이션 바와 비교하면서 개선점을 찾았습니다.

처음 기획과 달리 다양한 기능이 추가된 후에는 그것에 맞게 서비스의 콘텐츠들을 더 잘 보여줄 수 있는 레이아웃으로 메인 화면을 변경하였습니다.





이렇게 더 좋은 UI에 대해 고민하는 경험들이 쌓이면서 좋은 디자인과 사용성을 가진 UI를 구현해 낼 수 있는 프론트 엔드 개발자가 될 수 있다고 생각합니다.



많은 업무에 대해서 꼼꼼하게 문서화할 수 있습니다.

아래에 있는 문서는 인턴 과정 중 데이터 오류를 찾는 작업을 했을 때 문제를 분석하고 해결 방안을 정리한 것입니다. '항해 99' 부트 캠프에 참여하였을 때, 프로젝트의 중간, 최종 발표에 쓰이는 발표 자료 준비도 담당하였습니다. 팀원들이 작성한 각자 맡은 부분에 대해 정리한 글을 최대한 간략하고 핵심적으로 정리하였고 그 결과 프로젝트의 성과가 잘 나타나도록 할 수 있었습니다.

저의 이런 문서화 능력을 활용하여 협업 과정에서 좋은 시너지를 낼 수 있다고 생각합니다.



생산 -> 생산현황의 생산실적은 실적을 기준으로 하기 때문에 생산 -> 일별생산현황의 총생산수량과 데이터가 같아야 한다.
하지만 현재 데이터 값은 상이하다. 생산실적 = 437,717 총생산수량 = 2,636,438

```
SELECT sum(A.work_amt) '일별생산현황 총생산수량'
FROM TB_ProcJob A
WHERE A.account_mon = '202001'
AND A.is_master = 'Y'
```

코드를 가립니다.

base Console DEV

Output 일별생산현황 총생산수량.number


1	437717
---	--------

기존의 생산실적의 쿼리에
INNER JOIN TB_Progress B
ON (A.account_mon = B.account_mon AND B.is_master = 'Y')와
Where A.account_mon = '202001' AND '202001'을
추가했을 때 생산실적과 같은 데이터가 나온다.

// work_amt는 작업수량
// account_mon은 작지회계년월
// prog_no는 공정코드 // is_master는 주공정 여부

★ 기존의 총생산수량 쿼리에 inner join을 통해서 공정데이터베이스에 존재하는 공정코드를 가지고 있고 주공정이여야 하는 조건을 추가하고
원래 work_dt (작업일자)로 기간을 조건을 부여했던 것을 account_mon(작지회계년월)를 사용하면 생산실적의 데이터와 같은 값이
나온다. ★

* 생산실적은 리스트에서 천의 자리 부터의 숫자만 표시된다.
* 그래서 실제 리스트에 표기되는 생산실적의 데이터 값은 437이다. (리스트에 데이터 단위가 천(EA)라고 표시되어 있음)



중간 발표자료



최종 발표자료

😊 감사합니다!

📧 연락처: syk767@gmail.com

🐙 GitHub