5조 하이파이브 (정윤주, 이우혁, 조민혁, 소원호, 김영도, 박가인)

개발보고서

<1. 팀주제>

후보1) 설계-생산 통합 프로세스 모니터링 및 협업 강화 시스템 구축

- 설계와 생산을 구분한 메뉴를 통해 전체 프로세스를 쉽게 파악하고 협업을 촉진하는 플랫폼.

후보2) 실시간 공정 모니터링 및 커뮤니케이션 강화 플랫폼

- 공정 진행 상태를 실시간으로 확인하고, 메신저와 To-Do List 기능을 통해 팀 간 커뮤니케이션과 업무 효율성을 높이는 시스템.

후보3) 업무 효율 향상을 위한 공정 및 작업 관리 시스템

- 마이페이지 To-Do List와 메신저를 통해 업무 진행 상황을 관리하고, 부서 간 효율적인 협업을 가능하게 하는 시스템.

후보4) 공정 진행 상태 실시간 조회 및 사용자 맞춤형 업무 관리 시스템

- 공정 진행 상황을 실시간으로 모니터링하고, 사용자가 직접 업무를 관리할 수 있는 마이페이지 기능을 제공하는 시스템.

최종 주제)
" 중간 관리자 중심 설계-생산 통합 프로세스 모니터링 및 협업 강화 시스템 구축"

- 설계와 생산을 구분한 메뉴를 통해 설계-생산 전체 프로세스를 쉽게 파악 가능
- 공정 진행 상태를 실시간으로 모니터링하고, 사용자가 직접 업무를 지시 및 관리 가능
- 메신저와 투두리스트 기능을 통해 팀 간 커뮤니케이션과 업무 효율성을 높일 수 있음

사용 대상) 공정 현황을 관리 및 통솔 하는 센터장

<2. 팀주제 선정배경>

처음 주제 탐색 단계에서 저희 팀은 공정 모니터링 시스템에 대해 다양한 후보를 고려했습니다.

조율 1: 초기에는 데이터 보안에 중점을 두어, 각 부서가 자신의 정보만 확인할 수 있는 시스템을 구성했습니다. 예를 들어, 생산팀은 생산 관련 정보만, 설계팀은 설계 관련 정보만 볼 수 있도록 계획하였습니다. 이는 보안 측면에서 매우 중요한 부분이었지만, 멘토님의 조언으로 삼성중공업의 모니터링 웹인 '심포니'를 참고한 후 의견을 조율하게 되었습니다. 심포니는 전체 공정을 모니터링함으로써 일의 효율을 높이는 시스템이라는 조언을 듣고, 저희도 특정 부서 정보만을 보여주는 것이 아닌, 전체 공정을 아우르는 시스템으로 주제를 선정하게되었습니다. 이로 인해 사용자들이 전체적인 진행 상황을 직관적으로 파악할 수 있고, 부서 간의 협업을 강화할 수 있는 시스템으로 발전시켰습니다.

조율 2: 저희 시스템의 주 타겟은 생산 및 설계 부서의 중관 관리자들로, 이들은 공정 전체를 한눈에 파악하고 빠른 결정을 내려야 할 필요가 있습니다. 처음에는 모니터링 시스템 외 부가 기능에도 많은 비중을 두었습니다. 캘린더, 메신저, 투두리스트 등의 기능을 홈 화면에 배치하여 개인 업무의 효율성을 높이는 것에 초점을 맞추었으나, 고위 관리자들의 필요에 맞추기 위해 공정 모니터링이라는 주제에 집중해야 한다고 판단되었습니다. 이에 팀 내 의견을 조율한 결과, 공정 데이터 중심으로 시스템을 재구성하기로 하였습니다. 부가적인 기능들은 세부 페이지로 배치하여 주제의 명확성을 높이고자 했습니다.

조율 3: 마이페이지 항목에 들어갈 요소가 충분하지 않아 마이페이지 대신 홈 화면에 스케줄 관리와 체크리스트, 오늘의 할 일 기능을 배치하여 사용자가 전반적인 공정과 자신의 일정을 동시에 관리할 수 있도록 하였습니다. 특히 중간 관리자들이 전체 공정 상황을 쉽게 확인하면서도 개인 업무 조율이 가능하도록 설계되었습니다. 또한, 메신저 기능을 별도로 배치하여 부서 간의 실시간 소통을 강화하고, 빠른 결정을 내릴 수 있도록 지원하는 방향으로 시스템을 구성하였습니다.

최종적으로, 저희는 **설계 및 생산 전체 공정을 모니터링**할 수 있는 시스템을 중심으로 하고, 업무 효율성을 위한 부가 기능들은 홈 화면에서 보완적으로 활용하는 방향으로 주제를 확정하게 되었습니다. 이로 인해 중간 관리자들이 **공정 진행 상황을 실시간으로 파악**하고, **필요한 의사 결정**을 더욱 신속하게 내릴 수 있는 환경을 제공할 수 있을 것입니다.

<3. 프로젝트>

<3-1.프로젝트 내용>

저희 팀의 프로젝트는 설계 및 생산 통합 프로세스 모니터링 및 협업 강화 시스템 구축을 핵심으로 하고 있습니다. 홈화면, 생산 및 설계화면, 메신저 화면으로 구성되어 있습니다.

1. 홈화면

홈 화면에서는 특정 공정에만 집중하기보다는 선박별 건조 프로세스 조회 기능을 통해 전체적인 진행 상황을 직관적으로 파악할 수 있도록 설계되었습니다. 선박별로 설계-자재준비/가공-조립-의장-도장-탑재-진수-안벽작업-시운전-명명식 단계를 확인할 수 있습니다. 또한 자재별 적재 현황과 구역별 적재 현황을 확인할 수 있습니다. 이를 통해 사용자는 공정의 각 단계를 한눈에 확인할 수 있으며, 공정 관리의 효율성을 높일 수 있습니다.

우측에는 회의 및 중요 일정을 확인할 수 있는 캘린더와 체크 리스트를 배치하였으며, 사용자가 직접 할 일을 추가하고 수정 할 수 있는 기능을 구현했습니다. 이를 통해 사용자는 부서별 할 일 및 일정에 대해서 체계적으로 관리 할 수 있으며 공정 관리뿐만 아니라, 개인 업무의 효율성까지 높일 수 있도록 돕습니다.

- 자재별 적재 현황(케이블/베어링/후판)
- 구역별 적재 현황(A~E1,2,3)
- 선박별 건조 프로세스 조회
- Today's Schedule
- Today's CheckList
- Calendar

2. 로그인화면

회원가입 및 로그인 시스템은 보안성을 강화하고, 사용자와 관리자 계정 관리를 효율적으로 할 수 있도록 설계되었습니다. 사용자 인증, 세션 관리, 보안 강화, 로그아웃 기능 등을 통해 체계적인 사용자 관리와 안전한 웹 페이지의 이용이 가능합니다.

- **회원가입 기능 구현**

  사용자는 사번, 이름, 이메일, 비밀번호, 부서 등의 정보를 입력하여 회원가입을 할 수 있습니다. 가입된 사용자 정보는 서버에 안전하게 저장되며, 해싱 처리된 비밀번호를 통해 보안이 강화됩니다.
- **회원가입 후 로그인 기능**

  회원가입 후, 사용자는 자신의 이메일과 비밀번호로 로그인할 수 있으며, 로그인 성공 시 메인 웹페이지로 이동하게 됩니다. 로그인된 사용자만이 메인 페이지에 접근할 수 있습니다.
- **관리자 계정 등록**

  관리자 계정도 회원가입을 통해 등록되며, 일반 사용자와 차별화된 접근 권한을 가집니다. 관리자 계정의 비밀번호는 해싱 처리되어 저장되며, 아무나 접근할 수 없도록 보안 처리가 강화되었습니다.
- **세션 관리**

  로그인 성공 시, 서버로부터 JWT 토큰을 발급받아 로컬 스토리지에 저장됩니다. 이 토큰을 사용해 로그인 상태를 유지하며, 페이지를 새로고침해도 로그인 상태가 유지됩니다.
- **토큰 기반 인증 시스템 구축 (JWT)**

  모든 API 요청에 JWT 토큰이 포함되도록 구현되었습니다. 이를 통해 인증이 필요한 API에 안전하게 접근할 수 있으며, axios 인터셉터를 통해 자동으로 토큰이 추가됩니다.
- **보안 강화**
  - **HTTPS 적용**: 데이터 전송 시 HTTPS를 통해 암호화된 통신이 이루어집니다. SSL 인증서를 설정하여 보안을 강화하며, 특히 사용자가 입력한 민감한 정보를 안전하게 보호합니다.
  - **XSS 공격 방지**: 사용자로부터 입력받은 데이터를 저장하기 전에 검증을 통해 XSS 공격을 방지합니다. 이를 위해 입력 검증 라이브러리를 사용하여 모든 입력 데이터를 안전하게 처리합니다.
- **API 요청 인증**

  모든 API 요청에는 인증 토큰이 자동으로 포함됩니다. 이를 통해 인증이 필요한 API 엔드포인트에 안전하게 접근할 수 있으며, 보안이 강화되었습니다.

- **로그아웃 기능**

  메인 웹페이지 하단의 로그아웃 버튼을 클릭하면, 저장된 토큰이 제거되며 세션이 종료되고 초기 로그인 화면으로 돌아갑니다. 이를 통해 사용자는 안전하게 로그아웃 할 수 있습니다.

- **구현된 기능들을 통해 얻을 수 있는 보안성**

- **사용자 인증**

  사번과 비밀번호를 통해 로그인할 수 있으며, 실제 사용자 검증이 백엔드 서버에서 이루어집니다.

- **세션 관리**

  로그인 성공 시 JWT 토큰을 로컬 스토리지에 저장하고, 페이지 새로고침 후에도 로그인 상태를 유지합니다.

- **보안 강화**

  HTTPS와 XSS 방지 기능을 통해 사용자 데이터를 안전하게 보호하며, 해싱된 비밀번호를 저장하여 데이터베이스 보안을 강화합니다.

- **API 요청 인증**

  모든 API 요청에 자동으로 인증 토큰이 포함되어 보안을 강화하며, 인증이 필요한 API에 안전하게 접근할 수 있습니다.

- **로그아웃 기능**

  사용자가 안전하게 로그아웃할 수 있으며, 로그아웃 시 세션이 종료되고 저장된 토큰이 제거됩니다.

3. 설계 화면 (구조/의장 생산설계 DP현황)

각 공정별로 세부 사항을 확인할 수 있도록 설계와 생산으로 나누어 메뉴를 구현하였습니다. 이를 통해 생산 및 설계의 부서별로 필요한 정보를 명확하게 파악할 수 있고, 공정 진행 상황에 따른 대응을 더욱 세밀하게 할 수 있도록 했습니다.

설계화면은 구조설계와 의장설계로 나누어 시각화했습니다.

D/P&BOM 내역별(계획, 목의, Supp 제작, 의장 등)로 각 부서의 진행 상황을 확인할 수 있습니다. 또한 D/P&BOM 내역별 협력사 참여 비중을 알 수 있는 그래프, 부서별 일정 준수 여부 그래프를 통해 지연 건수를 한눈에 확인할 수 있도록 했습니다. 또한 월별 작업량 그래프 및 향후 12개월 작업량 예측 그래프로 작업량 추이를 파악할 수 있습니다.

- D/P&BOM 내역별 각 부서의 진행 상황
- D/P&BOM 내역별 협력사 참여 비중
- 부서별 일정 준수 여부
- 2021~2024년 월별 작업량
- 향후 12개월 간 작업량 예

4. 생산 화면 (용접불량률 현황)

항목별 용접 불량 정보를 조회할 수 있는 기능을 구현했습니다. 선종, 과, 업체, 사유코드, 용접기법별로 용접 불량률 분포와 불량 건수를 확인할 수 있습니다. 또한 선박코드 검색으로 선박별 불량 용접 데이터를 확인할 수 있으며, 불량 데이터를 추가 및 삭제하는 등 수정이 가능합니다. SHAP 분석을 통한 불량률 기여도 분석으로 변수별로 불량률에 미치는 영향을 확인할 수 있습니다. 이 수치가 클수록 불량률에 미치는 영향이 크다는 것을 알 수 있습니다.

- 선종, 과, 업체, 사유코드, 용접기법별 불량 정보 조회
- 선종별 불량률 분포
- 선종별 불량 건수
- 선박별 불량 용접 데이터 확인
- SHAP 분석을 통한 불량률 기여도 분석

5. 메신저 화면

관리자, 생산 및 설계 사원들과 소통할 수 있도록 메신저를 만들었습니다.

이 기능을 통해 친구추가 기능을 이용할 수 있습니다. 이를 통해 효율적인 소통 및 즉각적인 연락이 가능합니다.

그리고 친구삭제 기능도 사용 가능합니다.

결과적으로, 저희가 구현한 시스템은 사용자들에게 설계 및 생산 공정 모니터링과 업무 관리의 통합적 접근을 제공함으로써, 부서 간 협업을 강화하고 업무 처리의 효율성을 극대화할 것으로 기대됩니다.

- 사원 검색
- 친구 추가 및 삭제

<3-2.웹 페이지 구조>

1. 프로젝트 전체 구조

이 프로젝트는 크게 서버(server)와 클라이언트(client)로 구성된 풀스택 웹 페이지입니다. 서버는 Node.js와 Express를 기반으로 하고, 클라이언트는 React로 구현되었습니다.

1.1 클라이언트(client) 구조

- src
    - components: 다양한 페이지에서 재사용될 수 있는 컴포넌트들이 모여있는 디렉토리. ex) 모달, 차트, 로드 상태 시각화 등.
    - pages: 각각의 페이지 컴포넌트가 위치. 각 페이지는 필요한 컴포넌트를 사용하여 독립된 화면을 렌더링.
    - services: 서버와 통신하는 API 호출 관련 파일들. 각 기능별로 API를 처리.
    - utils: 유틸리티 함수들이 모여있는 디렉토리.

1.2 서버(server) 구조

    - config: 데이터베이스나 서버 설정 관련 정보.
    - controllers: 각 라우트에서 호출되는 비즈니스 로직 처리. 데이터베이스 작업이나 모델과의 상호작용을 담당.
    - db: SQL 스크립트와 데이터베이스 설정 관련 파일들.
    - middlewares: 라우트로 들어오는 요청을 가로채어 처리하는 중간 처리 로직. 주로 인증 관련 작업 수행.
    - models: 데이터베이스의 테이블 구조를 정의하는 파일들. 주로 Mongoose나 ORM을 사용.
    - routes: 각 페이지와 관련된 라우트 정의. API 엔드포인트를 처리하는 역할.
    - scripts: 서버에서 실행할 수 있는 스크립트들. 데이터베이스 초기화나 유틸리티 스크립트.

2. 페이지별 구성 및 작동 방식

2.1. 로그인 페이지 (index.js)

- 파일 경로: LoginPage/index.js
- 관련 컴포넌트: RegisterModal, 사용자가 회원가입을 할 수 있는 모달 창을 구현한 React 컴포넌트입니다.
- API 관련 파일:
    - api.js (클라이언트 서비스): Axios를 사용하여 서버와 통신하는 기본 API 설정을 정의합니다.

- 작동 방식:
    - 사용자는 이메일과 비밀번호를 입력하고, 로그인 폼을 제출합니다.
    - 폼 제출 시 서버의 /auth/login 엔드포인트로 POST 요청을 보내, 사용자 정보를 확인하고 JWT 토큰을 반환받습니다.
    - 로그인에 성공하면 토큰을 localStorage에 저장하고 메인 페이지로 이동합니다.
    - 서버 관련 파일:
        - /routes/auth.js (라우트): 로그인 관련 라우트를 처리합니다.
        - /middlewares/auth.js (미들웨어): 인증 관련 미들웨어. JWT 토큰을 검증하여 사용자 세션을 유지.
        - authController.js (컨트롤러): 사용자 인증 로직을 처리하여, 토큰을 생성합니다.

2. 메인 페이지 (App.js)

- 파일 경로: App.js
- 관련 컴포넌트:
    - Header: 메인 페이지 상단의 헤더 컴포넌트, 사용자 로그아웃 기능 포함.
    - LoadStatusChart: 특정 위치의 작업 상태를 시각화하는 차트 컴포넌트.
    - Summary: 선택된 위치에 따른 요약 데이터를 보여주는 컴포넌트.
    - TodaySchedule, To-do List, Calendar: 오늘의 일정 및 할 일 목록, 달력을 관리.
    - ShipbuildingStatus: 조선 작업의 진행 상태를 보여주는 컴포넌트.
- API 관련 파일:
    - shipbuildingService.js (클라이언트 서비스): 조선 공정 데이터를 가져오는 API 요청을 처리합니다.

- 작동 방식:

  페이지 로드 시 각 컴포넌트가 데이터를 로드하고 시각화하며, 사용자가 페이지 내에서 정보를 확인합니다.

    - 서버 관련 파일:
        - cargo.js (라우트): 물류 및 창고 관리와 관련된 API 요청을 처리합니다.
        - shipbuilding.js (라우트): 조선 공정 데이터 관련 API 요청을 처리합니다.
        - todo.js (라우트): 할 일 관리(TODO) 관련 API 요청을 처리합니다.
        - calendar.js (라우트): 일정 관련 API 요청을 처리합니다
        - cargoController.js (컨트롤러): 특정 저장 위치와 자재의 저장량 및 창고의 남은 용량을 계산하고, 저장 위치와 자재 목록을 조회합니다.

- shipbuildingController.js (컨트롤러): 특정 선박 ID에 대한 최신 공정 데이터를 조회하여 반환합니다
- todoController.js (컨트롤러): 할 일 목록 조회, 추가, 삭제, 순서 변경, 업데이트 등의 기능을 제공합니다.
- calendarController.js (컨트롤러): 일정 목록 조회, 일정 추가, 삭제, 업데이트, 오늘 일정 조회.

3. 생산 페이지 (DefectRate.js)

- 파일 경로: DefectRate_Page/DefectRate.js
- 관련 컴포넌트:
  - DefectRateMenu.js: 결함 비율 페이지에서 데이터를 조회할 메뉴 또는 필터 기능을 제공합니다. 사용자는 특정 카테고리나 데이터를 선택할 수 있습니다.
  - DefectRatePieChart.js: 결함 데이터의 비율을 원형 차트 형태로 시각화합니다. 전체 결함 비율을 시각적으로 쉽게 이해할 수 있게 해줍니다.
  - DefectRateBarChart.js: 결함 비율 데이터를 막대 차트 형태로 시각화합니다. 선택된 카테고리나 필터에 따른 결함 비율 분포를 보여줍니다.
  - DefectRateGreenSection.js: 결함 비율의 세부 데이터를 보여주는 섹션 중 하나입니다. 주로 선택된 프로젝트의 결함 데이터를 상세히 보여줍니다.
  - DefectRateBlueSection.js: SHAP 분석 결과를 시각화하는 섹션입니다. 이 섹션은 각 변수들이 결함 데이터에 미치는 영향을 분석한 결과를 차트로 나타냅니다.
- 작동 방식:
  - defectRateService.js에서 서버로 데이터를 요청하고, 받아온 결함 데이터를 다양한 차트 형태로 시각화합니다.
- 서버 관련 파일:
  - defectRateRoutes.js (라우트): 생산 결함 관련 데이터를 처리하는 라우트.
  - defectRateController.js (컨트롤러): 결함 데이터를 처리하고 클라이언트에 제공.

4. 설계 페이지 (DesignStructure.js)

- 파일 경로: DesignStructure_Page/DesignStructure.js
- 관련 컴포넌트:
  - VisualizationComponent.js: 부서별 작업 비율을 막대 차트로 시각화하고, 각 부서의 비율을 강조하여 시각적으로 보여줍니다
  - PartnerParticipationVisualization.js: D3.js를 사용하여 막대 차트로 협력사별 작업 비율을 시각화하고, 각 협력사의 기여도를 한눈에 파악할 수 있도록 합니다.
  - ScheduleComplianceVisualization.js: 각 부서의 준수 및 지연 데이터를 차트로 시각화하고, 지연 비율을 강조하여 사용자에게 명확하게 전달합니다.

    - MonthlyWorkAmountVisualization.js: 월별 작업량 데이터를 선형 그래프로 시각화하여 연간 작업량의 변동을 보여줍니다
    - FutureWorkPredictionVisualization.js: Holt-Winters 모델 활용해 향후 12개월 동안의 작업량을 예측하여 선형 차트로 시각화합니다.
- API 관련 파일:
    - dpbomDepartmentService.js (클라이언트 서비스): 서버와 통신하여 설계 데이터를 가져오는 API 호출을 담당합니다(DesignStructure).

- 작동 방식:
    - 사용자가 설계 타입을 선택하고, 해당 데이터(구조설계 또는 의장설계)를 서버에서 가져와 시각화합니다.
    - 서버 관련 파일:
        - dpbomDepartment.js (라우트): 설계 데이터 관련 API 라우트.
        - dpbomDepartmentController.js (컨트롤러): 설계 데이터 로직 처리.

5. 채팅 모달 (ChatModal.js)

- 파일 경로: components/src/ChatModal.js
- 관련 컴포넌트: ChatModal.js => 사용자가 친구와 실시간 채팅을 할 수 있는 모달창을 구현한 React 컴포넌트입니다. 주요 기능은 사용자와 친구 목록을 표시하고, 선택한 친구와의 대화를 지원합니다.
- API 관련 파일: chatService.js (클라이언트 서비스) => 클라이언트에서 API 호출을 통해 친구 추가, 삭제, 메시지 전송 등을 관리.
- 작동 방식:
    - 사용자는 친구 목록에서 대화 상대를 선택하고, 실시간으로 채팅 메시지를 주고받습니다.
    - Socket.io를 통해 실시간 메시지 전송 기능을 구현하며, 서버의 /messages API로 메시지를 주고받습니다.
    - 서버 관련 파일:
        - chat.js (라우트): 사용자, 친구, 메시지 관련 API를 처리.

- 이 프로젝트의 디렉토리 구조와 페이지별 컴포넌트 및 서버 간의 상호작용을 통해 각 기능이 분리되어 있으며, MVC 패턴에 기반한 구조로 명확하게 역할을 나누어 개발되었습니다.

<3-3.활용방안 및 기대 효과>

활용 방안:

1. **실시간 공정 모니터링**: 각 부서에서 실시간으로 **선박별 건조 진행 상황**을 확인할 수 있으므로, 공정 지연이나 문제 발생 시 즉각적으로 대응할 수 있습니다. 특히 고위 관리자들은 전체 공정을 한눈에 파악할 수 있어 빠른 의사 결정을 내릴 수 있습니다.
2. **자재 관리 효율화**: 자재별 및 구역별 적재 현황을 확인할 수 있는 기능을 통해 자재 흐름을 체계적으로 관리하고, 자재 부족이나 적재 문제를 사전에 파악하여 해결할 수 있습니다.
3. **부서 간 협업 강화**: **메신저 기능**을 통해 부서 간의 실시간 소통이 가능해지면서, 공정 진행 중 발생할 수 있는 문제들을 빠르게 공유하고 해결할 수 있습니다. 이를 통해 부서 간 협업이 강화되고, 공정 전체의 효율이 높아집니다.
4. **일정 관리 및 업무 조율**: 고위 관리자들은 공정 모니터링과 더불어 자신의 업무 일정을 관리할 수 있는 **스케줄 관리와 체크리스트 기능**을 통해 업무 효율성을 극대화할 수 있습니다.

기대 효과:

1. **생산성 향상**: 공정 상황을 실시간으로 모니터링하고, 자재와 구역별 상황을 즉각 확인할 수 있어, 공정 지연이나 문제를 신속하게 해결할 수 있습니다. 이로 인해 전체 생산성이 향상될 것으로 기대됩니다.
2. **리스크 최소화**: 실시간 데이터를 기반으로 **공정 리스크를 사전에 파악**하여 빠르게 대응할 수 있으므로, 공정 실패나 납기 지연을 최소화할 수 있습니다.
3. **업무 효율성 증대**: 부가 기능들을 통해 관리자들이 개인 업무를 효율적으로 관리할 수 있으며, 부서 간 소통이 원활해짐에 따라 **업무 처리 속도**와 **정확도**가 향상될 것입니다.
4. **비용 절감**: 공정 관리 효율화와 신속한 의사 결정으로 인해, 불필요한 자재 낭비나 시간 손실을 줄이고 **비용 절감 효과**를 기대할 수 있습니다.

<4.각 멘토별 지원내역>

1. 이근우 프로님

1.1 홈페이지 화면 수정

기존에 제작하던 홈페이지의 디자인이, 프로젝트에서 요구하는 모니터링 홈페이지와는 거리가 멀다는 피드백을 받았습니다. 이후 저희 조에서는 SNS스러운 디자인 대신에 전문성과 유연성을 갖춘 페이지를 만들기 위해 노력하였습니다.

2. 이동현 프로님

2.1. 평가 기준
프로님이 생각하시는 현업에서 모니터링시 중요한게 무엇인가요?

- 학습한 거 기반으로 잘 구현 했는가
- 자료, 발표 위주 확인 할것
- 디자인적+ 의도한 기능이 잘 구현 되었는가
- 왜 이런 화면을 만들었고, 이러한 기능이 필요한 이유에 대해서 설명해야함
- ex) 목적 명시할것, ~상황에서 ~에 쓰일 기능 ~를 제공해 줄 것이다.

2.2. 프로님이 생각하시는 현업에서 모니터링시 중요한게 무엇인가요?

1. 타겟팅

- 타겟팅을 할때 실적 받는 사람인지 실적을 내야하는 사람인지에 따라 하는 일이 다를 것인데,
- 예를 들어 출도를 수정해야하는 사람이 있을 거고, 수정에 따라 정보를 확인해서 움직여야하는 사람이 있을 텐데 그들의 역할에 따라 기능이 구현 되야함
- 우리 같은 경우는 그룹장이므로 부서별 비교라던지, 현장 상황을 한눈에 확인 가능한 기능을 위주로 구현하는 것이 좋음

2. 화면 구성

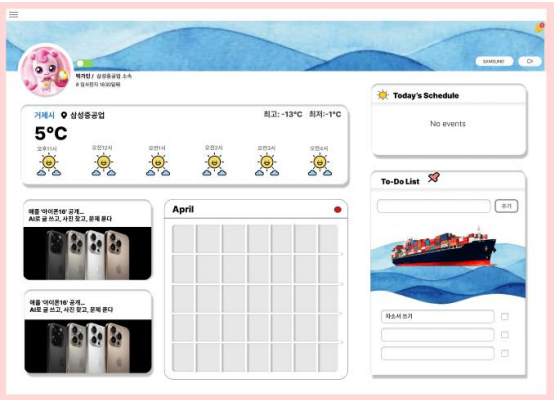
- 처음 페이지에 들어갔을때 직관적으로 각 기능에 대해 알 수 있게끔 하는 것이 젤 중요
- 실제로는 p1,p2,p3 까지만 있는 것이 아닌 20개 이상이므로 잘 고려할것
- 화면에 빈 느낌이 없게끔
- 클릭해야되는 기능같은거 화면보고 젤 첨에 뭘 해야하고 어떤걸 사용해야하고 원하는 정보가 어디 있는지 잘 알 수 있게끔 해야함
- 요약적하자면, 직관적으로 만들어야함

3. 윤가림프로님

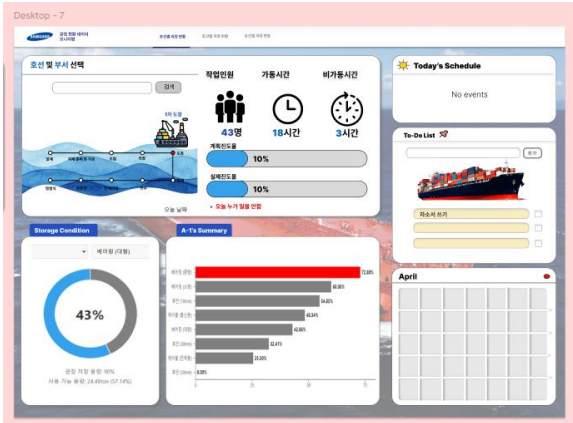
3.1 결과보고서 수정

멘토님께서 가동성 확인 후, 필요한 추가 내용과 중요한 사항을 피드백 해주셨으며,, **핵심 기능의 작동과 내용의 명확성을 강조**

<5.프로젝트 수정사항>



->



공정현황 메인페이지에 뉴스와 날씨를 넣었으나, 이근우 프로님께 미니 홈페이지처럼 보인다는 이야기를 들었습니다. 따라서 저희 조는 공정현황 모니터링 웹 페이지로 보이지 않는다고 판단하여 이를 개선하기 위해 홈 화면에 공정현황을 확실하게 보여줄 수 있는 데이터를 배치하였습니다.

그 결과 아래와 같은 홈 화면으로 변경하였습니다.
선박별 건조 프로세스를 조회할 수 있도록 하고, 자재별 적재 현황과 구역별 적재 현황을 알 수 있도록 수정하였습니다.