포팅매뉴얼

진행 상태	시작 전
태그	

1. 기술 스택 & 개발 환경

[사용 도구]

- 이슈 관리: JIRA
- 형상 관리: GitLab
- 커뮤니케이션: Mattermost, Notion
- 디자인: Figma, Canva, Miricanvas
- UCC: Movavi, Vrew
- CI/CD: EC2, Docker, Jenkins

[개발 환경]

- Front-end
  - Node.js 20
  - React 19.1.0
  - Zustand 5.0.6
  - Tailwind 4.1.11
  - PWA
  - Prettier 3.6.2
  - VS Code 1.103.1
- Back-end
  - JDK 21.0.7 LTS

    - Spring Boot 3.5.4 (Web, Security, WebSocket)
    - MyBatis 3.0.3 + MySQL 8.x (Connector/J)
    - MQTT (Eclipse Paho 1.2.5)
    - AWS SDK for S3 v2.24.12 (presigned URL)
    - Jackson (JSR-310)
    - Lombok 1.18.30, DevTools
    - BouncyCastle (PEM/TLS)
    - IntelliJ: IntelliJ IDEA 2023.3.8 (Ultimate Edition)
    - Cursor: 1.4.5
- DB
    - MySQL 8.0.37
- Embedded
    - Ubuntu 22.04 LTS
    - ROS 2 Humble
    - Python 3.10.11
    - Cartograper
    - Navigation 2
- Infra
    - Docker 28.3.2 (Docker Engine - Community)
    - AWS EC2 22.04.4 LTS
    - Jenkins jenkins:lts
    - Nginx nginx/1.18.0 (Ubuntu)

[외부 서비스]

- firebase 12.1.0

[gitignore]

- 공통

```
# Created by https://www.toptal.com/developers/gitignore/api/windows,macos,intellij,visualstudiocode,eclipse,java,react
# Edit at https://www.toptal.com/developers/gitignore?templates=windows,macos,intellij,visualstudiocode,eclipse,java,react

### Eclipse ###
.metadata
bin/
tmp/
*.tmp
*.bak
*.swp
*~.nib
local.properties
.settings/
.loadpath
.recommenders

# External tool builders
.externalToolBuilders/

# Locally stored "Eclipse launch configurations"
*.launch

# PyDev specific (Python IDE for Eclipse)
*.pydevproject

# CDT-specific (C/C++ Development Tooling)
.cproject

# CDT- autotools
.autotools

# Java annotation processor (APT)
.factorypath

# PDT-specific (PHP Development Tools)
.buildpath
```

```
# sbteclipse plugin
.target

# Tern plugin
.tern-project

# TeXlipse plugin
.texlipse

# STS (Spring Tool Suite)
.springBeans

# Code Recommenders
.recommenders/

# Annotation Processing
.apt_generated/
.apt_generated_test/

# Scala IDE specific (Scala & Java development for Eclipse)
.cache-main
.scala_dependencies
.worksheet

# Uncomment this line if you wish to ignore the project description file.
# Typically, this file would be tracked if it contains build/dependency config
urations:
#.project

### Eclipse Patch ###
# Spring Boot Tooling
.sts4-cache/

### Intellij ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyChar
m, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/20654
```

```
4839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# Gradle and Maven with auto-import
# When using Gradle or Maven with auto-import, you should exclude modul
e files,
# since they will be recreated, and may cause churn.  Uncomment if using
# auto-import.
# .idea/artifacts
# .idea/compiler.xml
# .idea/jarRepositories.xml
# .idea/modules.xml
# .idea/*.iml
```

```
# .idea/modules
# *.iml
# *.ipr

# CMake
cmake-build-*/

# Mongo Explorer plugin
.idea/**/mongoSettings.xml

# File-based project format
*.iws

# IntelliJ
out/

# mpeltonen/sbt-idea plugin
.idea_modules/

# JIRA plugin
atlassian-ide-plugin.xml

# Cursive Clojure plugin
.idea/replstate.xml

# SonarLint plugin
.idea/sonarlint/

# Crashlytics plugin (for Android Studio and IntelliJ)
com_crashlytics_export_strings.xml
crashlytics.properties
crashlytics-build.properties
fabric.properties

# Editor-based Rest Client
.idea/httpRequests

# Android studio 3.1+ serialized cache file
```

```
.idea/caches/build_file_checksums.ser

### Intellij Patch ###
# Comment Reason: https://github.com/joeblau/gitignore.io/issues/186#issuecomment-215987721

# *.iml
# modules.xml
# .idea/misc.xml
# *.ipr

# Sonarlint plugin
# https://plugins.jetbrains.com/plugin/7973-sonarlint
.idea/**/sonarlint/

# SonarQube Plugin
# https://plugins.jetbrains.com/plugin/7238-sonarqube-community-plugin
.idea/**/sonarIssues.xml

# Markdown Navigator plugin
# https://plugins.jetbrains.com/plugin/7896-markdown-navigator-enhanced
.idea/**/markdown-navigator.xml
.idea/**/markdown-navigator-enh.xml
.idea/**/markdown-navigator/

# Cache file creation bug
# See https://youtrack.jetbrains.com/issue/JBR-2257
.idea/$CACHE_FILE$

# CodeStream plugin
# https://plugins.jetbrains.com/plugin/12206-codestream
.idea/codestream.xml

# Azure Toolkit for IntelliJ plugin
# https://plugins.jetbrains.com/plugin/8053-azure-toolkit-for-intellij
.idea/**/azureSettings.xml
```

```
### Java ###
# Compiled class file
*.class

# Log file
*.log

# BlueJ files
*.ctxt

# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar

# virtual machine crash logs, see http://www.java.com/en/download/help/e
rror_hotspot.xml
hs_err_pid*
replay_pid*

### macOS ###
# General
.DS_Store
.AppleDouble
.LSOverride

# Icon must end with two \r
Icon


# Thumbnails
```

```
._*

# Files that might appear in the root of a volume
.DocumentRevisions-V100
.fseventsd
.Spotlight-V100
.TemporaryItems
.Trashes
.VolumeIcon.icns
.com.apple.timemachine.donotpresent

# Directories potentially created on remote AFP share
.AppleDB
.AppleDesktop
Network Trash Folder
Temporary Items
.apdisk

### macOS Patch ###
# iCloud generated files
*.icloud

### react ###
.DS_*
logs
**/*.backup.*
**/*.back.*

node_modules
bower_components

*.sublime*

psd
thumb
sketch

### VisualStudioCode ###
```

```
.vscode/*
!.vscode/settings.json
!.vscode/tasks.json
!.vscode/launch.json
!.vscode/extensions.json
!.vscode/*.code-snippets

# Local History for Visual Studio Code
.history/

# Built Visual Studio Code Extensions
*.vsix

### VisualStudioCode Patch ###
# Ignore all local history of files
.history
.ionide

### Windows ###
# Windows thumbnail cache files
Thumbs.db
Thumbs.db:encryptable
ehthumbs.db
ehthumbs_vista.db

# Dump file
*.stackdump

# Folder config file
[Dd]esktop.ini

# Recycle Bin used on file shares
$RECYCLE.BIN/

# Windows Installer files
*.cab
*.msi
*.msix
```

```
*.msm
*.msp

# Windows shortcuts
*.lnk

HELP.md
target/
!**/src/main/**/target/
!**/src/test/**/target/

### STS ###
.apt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache

### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr

### NetBeans ###
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/
build/
!**/src/main/**/build/
!**/src/test/**/build/

### VS Code ###
.vscode/
```

```
### gitlab
.gitlab/

# IDE 설정
.server/.idea/
.server/.vscode/

# 환경변수 파일
.env
application.properties

# 임베디드 gitignore
# ROS 2 Workspace Files
robot_ws/build/
robot_ws/install/
robot_ws/log/

# 지도 파일
robot_ws/maps/

# 파이썬 캐시
**/__pycache__/
*.pyc

# 기타
*~
*.swp
.vscode/

# End of https://www.toptal.com/developers/gitignore/api/windows,macos,
intellij,visualstudiocode,eclipse,java,react

# Created by https://www.toptal.com/developers/gitignore/api/python
# Edit at https://www.toptal.com/developers/gitignore?templates=python

### Python ###
# Byte-compiled / optimized / DLL files
```

```
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# PyInstaller
#  Usually these files are written by a python script from a template
#  before PyInstaller builds the exe, so as to inject date/other infos into it.
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt

# Unit test / coverage reports
```

```
htmlcov/
.tox/
.nox/
.coverage
.coverage.*
.cache
nosetests.xml
coverage.xml
*.cover
*.py,cover
.hypothesis/
.pytest_cache/
cover/

# Translations
*.mo
*.pot

# Django stuff:
*.log
local_settings.py
db.sqlite3
db.sqlite3-journal

# Flask stuff:
instance/
.webassets-cache

# Scrapy stuff:
.scrapy

# Sphinx documentation
docs/_build/

# PyBuilder
.pybuilder/
target/
```

```
# Jupyter Notebook
.ipynb_checkpoints

# IPython
profile_default/
ipython_config.py

# pyenv
#   For a library or package, you might want to ignore these files since the c
ode is
#   intended to run in multiple environments; otherwise, check them in:
# .python-version

# pipenv
#   According to pypa/pipenv#598, it is recommended to include Pipfile.loc
k in version control.
#   However, in case of collaboration, if having platform-specific dependen
cies or dependencies
#   having no cross-platform support, pipenv may install dependencies that
don't work, or not
#   install all needed dependencies.
#Pipfile.lock

# poetry
#   Similar to Pipfile.lock, it is generally recommended to include poetry.loc
k in version control.
#   This is especially recommended for binary packages to ensure reproduc
ibility, and is more
#   commonly ignored for libraries.
#   https://python-poetry.org/docs/basic-usage/#commit-your-poetrylock-f
ile-to-version-control
#poetry.lock

# pdm
#   Similar to Pipfile.lock, it is generally recommended to include pdm.lock i
n version control.
#pdm.lock
#   pdm stores project-wide configurations in .pdm.toml, but it is recommen
```

```
ded to not include it
#   in version control.
#   https://pdm.fming.dev/#use-with-ide
.pdm.toml

# PEP 582; used by e.g. github.com/David-OConnor/pyflow and github.co
m/pdm-project/pdm
__pypackages__/

# Celery stuff
celerybeat-schedule
celerybeat.pid

# SageMath parsed files
*.sage.py

# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/

# Spyder project settings
.spyderproject
.spyproject

# Rope project settings
.ropeproject

# mkdocs documentation
/site

# mypy
.mypy_cache/
.dmypy.json
```

```
dmypy.json

# Pyre type checker
.pyre/

# pytype static type analyzer
.pytype/

# Cython debug symbols
cython_debug/

# PyCharm
#  JetBrains specific template is maintained in a separate JetBrains.gitignore that can
#  be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
#  and can be added to the global gitignore or merged into this file.  For a more nuclear
#  option (not recommended) you can uncomment the following to ignore the entire idea folder.
#.idea/

### Python Patch ###
# Poetry local configuration file - https://python-poetry.org/docs/configuration/#local-configuration
poetry.toml

# ruff
.ruff_cache/

# LSP config files
pyrightconfig.json

# firebase 환경변수
firebase-key.json

# End of https://www.toptal.com/developers/gitignore/api/python
```

- Back-end

```
#####################################
# Build Outputs
#####################################
target/
build/
out/
test-output/
coverage/
jacoco.exec

#####################################
# Maven & Wrapper
#####################################
.mvn/wrapper/maven-wrapper.jar
HELP.md

#####################################
# IDE Settings
#####################################
# IntelliJ / Android Studio
.idea/
*.iml
*.iws
*.ipr

# VS Code
.vscode/

# Eclipse / STS
.classpath
.project
.settings/
.springBeans
.sts4-cache
.apt_generated
.factorypath
```

```
# NetBeans
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/

#########################################
# Environment & Secrets
#########################################
.env
.env.*
application.properties
application-*.yml
application-*.properties

#########################################
# System Files & Temp
#########################################
.DS_Store
Thumbs.db
*.log
*.tmp
*.bak
*.swp
*.pid
replay_pid*
hs_err_pid*

#########################################
# Exceptions (Include specific target/build if needed)
#########################################
!**/src/main/**/target/
!**/src/test/**/target/
!**/src/main/**/build/
!**/src/test/**/build/
```

[환경 변수]

- .env

```
# 로컬 개발용
VITE_REACT_APP_SPRING_BASE_URL=http://localhost:{로컬 서버 포트}

# 배포환경 개발용
VITE_REACT_APP_SPRING_BASE_URL={배포 주소}
```

- application.properties

```
#Server
server.port={서버용 포트}
spring.application.name=areuhot
server.servlet.context-path=/api
server.ssl.enabled=false

#DB
spring.datasource.url=jdbc:mysql://{도메인}:{DB용 포트}/{DB 이름}?serverTi
mezone=Asia/Seoul&useSSL=false
spring.datasource.username={DB 아이디}
spring.datasource.password={DB 패스워드}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#MyBatis
mybatis.mapper-locations=classpath:mapper/**/*.xml
mybatis.type-aliases-package=kr.kro.areuhot
mybatis.configuration.map-underscore-to-camel-case=true
mybatis.configuration.default-enum-type-handler=org.apache.ibatis.type.E
numTypeHandler

#Logging
logging.level.root=INFO
logging.level.org.springframework.web=DEBUG
logging.level.kr.kro.areuhot=DEBUG

#MQTT Logging Configuration
logging.level.org.eclipse.paho.client.mqttv3=WARN
```

```
logging.level.kr.kro.areuhot.alert=DEBUG

#AWS S3 Configuration
aws.accessKey={AWS Access key}
aws.secretKey={AWS Secret key}
aws.bucket.name={AWS buckert name}
aws.bucket.region={AWS region}
aws.bucket.prefix={AWS prefix}

#MQTT Configuration
mqtt.endpoint={MQTT endpoint}
mqtt.topic={MQTT topic}
mqtt.topic.location={MQTT location}
mqtt.port={MQTT port}
mqtt.client.id={MQTT client id}

#MQTT Advanced Configuration
mqtt.connection.timeout=30
mqtt.keepalive.interval=60
mqtt.max.inflight=1000
mqtt.reconnect.max.attempts=5
mqtt.reconnect.delay.ms=5000

#MQTT Certificate Configuration
mqtt.certificate.ca=AmazonRootCA1.pem
mqtt.certificate.client=certificate.pem.crt
mqtt.certificate.private=private.pem.key

#MQTT QoS Configuration
mqtt.qos=1

# firebase
firebase.key=classpath:{firebase-key.json 파일 경로}
```

2. 빌드 및 배포

개발 환경 (로컬 빌드)

- 프론트엔드

```
cd frontend

# 의존성 설치
npm install

# 프로젝트 실행
npm run dev
```

- 백엔드

```
cd server

# 의존성 설치
./mvnw clean package

# 프로젝트 실행
./mvnw spring-boot:run
```

- 임베디드

```
# 0. ROS2 개발 환경 구축
https://docs.ros.org/en/humble/Installation.html#binary-packages

# 1. 워크스페이스 생성
mkdir -p ~/ros2_ws/src
cd ~/ros2_ws/src

# 2. 프로젝트 클론
git clone <this-repo-url> robot
cd ..

# 3. 빌드 및 환경설정
colcon build --symlink-install
source /opt/ros/humble/setup.bash
source install/setup.bash
```

```
# 4. SLAM 실행 (Cartographer)
ros2 launch my_cartographer now_slam_test.launch.py

# 5. 네비게이션 실행 (Nav2)
ros2 launch my_cartographer bringup_ackermann_nav2.launch.py

# 6. 키보드 조작 (Teleop)
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

배포시 빌드

- Jenkins pipeline

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                // 작업 공간을 깨끗하게 정리한다(선택 권장사항)
                cleanWs()

                git branch: 'master',
                    url: 'https://lab.ssafy.com/s13-webmobile3-sub1/S13P11A202.gi
t',
                    credentialsId: 'gitlab-token'
            }
            post {
                success {
                    sh 'echo "Successfully Cloned Repository"'
                }
                failure {
                    sh 'echo "Fail Cloned Repository"'
                }
            }
        } // stage('Checkout')

        stage('Load Back Properties'){
```

```
        steps {
            withCredentials([file(credentialsId: '{환경변수 등록ID}', variable:'{Pipeline에서 사용할 변수}'), file(credentialsId: '{환경변수 등록ID}', variable:'{Pipeline에서 사용할 변수}')]){
                script{
                    sh 'mkdir -p server/src/main/resources' // resources 디렉토리가 없으면 만들어준다
                    sh 'cp ${Pipe에서 사용할 변수} server/src/main/resources/application.properties'
                    sh 'cp ${Pipe에서 사용할 변수} server/src/main/resources/firebase-key.json'
                    sh 'ls server/src/main/resources'
                }
            }


        }
    } // stage('Load Back Properties')

    stage('Load Front Properties'){
        steps {
            withCredentials([file(credentialsId: '{환경변수 등록ID}', variable:'{Pipe에서 사용할 변수}')]){
                script{
                    sh 'mkdir -p frontend'
                    sh 'cp ${Pipe에서 사용할 변수} frontend/.env'
                    sh 'ls frontend'
                }
            }
        }
    } // stage('Load Front Properties')

    stage('Deploy SLAM Server') {
        steps {
            withCredentials([file(credentialsId: '{환경변수 등록ID}', variable: '{Pipeline에서 사용할 변수}')]) {
                sh """
                    # slam-mqtt-server 디렉토리로 이동
```

```
                cd slam-mqtt-server

                # 시크릿 env 복사
                cp \${Pipeline에서 사용할 변수} .env

                # 이미지 빌드
                docker build -t slam-mqtt-server .

                # 기존 컨테이너가 있다면 삭제
                docker rm -f slam-server || true

                # 새 컨테이너 실행
                docker run -d -p 5000:5000 \
                  --env-file .env \
                  --name slam-server slam-mqtt-server
            """
        }
    }
} // stage('Deploy SLAM Server')

stage('test') {
    steps {
        dir('server') {
            sh """
                chmod +x mvnw
                ./mvnw clean install

                ls target
                """
        }
    }
} // stage('test')

stage('Clean up old containers') {
    steps {
        sh 'docker-compose down' // -v 옵션은 연결된 볼륨까지 삭제
    }
}
```

```
        stage('delpoy docker-compose') {
            steps {
                // docker-compose 파일이 있는 경로에서 실행 (현재 프로젝트 루트에 있
음)
                // --force-recreate : 이미지가 변경되지 않았더라도 컨테이너를 다시 만
들도록 보장
                sh 'docker-compose build --no-cache'
                sh 'docker-compose up -d --build --force-recreate'
            }
        } // stage('delpoy docker-compose')

    } //stages

} // pipeline
```

- docker-compose.yml (root/docker-compose.yml)

```
version: "3.8"

services:
  backend:
    # 빌드 이미지의 이름을 명시적으로 지정
    image: areuhotspring
    build:
      context: ./server
    container_name: areuhot-spring-container
    ports:
      - "8080:8080"
    # 사용자가 중지하지 않는 이상 자동 재시작
    restart: unless-stopped
    volumes:
      - backend_data:/app/data # .lck 파일 저장 경로
    networks:
      - app-network
    command: sh -c "chmod -R 777 /app/data && sleep 1 && java -Duser.tim
ezone=Asia/Seoul -jar app.jar"
```

```
  frontend:
    image: areuhotreact
    build:
      context: ./frontend
    container_name: areuhot-react-container
    ports:
      - "5173:80"
    depends_on:
      - backend
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

volumes:
  backend_data:
```

- Front-end Dockerfile (root/frontend/Dockerfile)

```
FROM node:20 AS build

WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
RUN npm run build

FROM nginx:stable-alpine
COPY --from=build /app/dist /usr/share/nginx/html
RUN rm /etc/nginx/conf.d/default.conf
COPY nginx/nginx.conf /etc/nginx/conf.d
EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

- Back-end Dockerfile (root/server/Dockerfile)

```
FROM eclipse-temurin:21-jdk AS build

WORKDIR /app
COPY . .
RUN chmod +x mvnw && ./mvnw clean package -DskipTests

FROM eclipse-temurin:21-jdk
WORKDIR /app
COPY --from=build /app/target/*.jar app.jar

# .lck 저장용 디렉토리 생성 및 권한 부여
RUN mkdir -p /app/data && chmod 777 /app/data

ENTRYPOINT ["java", "-Duser.timezone=Asia/Seoul", "-jar", "app.jar"]
```

- Nginx 설정 파일 (root/frontend/nginx/nginx.conf)

```
# 기존 설정내용
# react-dockerizing/nginx/nginx.conf

server {
    listen 80;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    location / {
        # root를 /usr/share/nginx/html 을 바라보게 했으므로(Dockerfile 참고)
        # 해당 경로 아래에 배포해주면 됨
        root   /usr/share/nginx/html;
        index  index.html index.htm;
        try_files $uri $uri/ /index.html;
    }
}
```

기타

PWA

- Firebase : Firebase의 FCM (Firebase Cloud Messaging)을 이용한 PWA 푸시알림 구현

```
# 구조
root/
├── server/
│   └── src/main/java/kr/kro/areuhot/
│       ├── config/
│       │   ├── AsyncConfig.java           | 비동기 처리 설정
│       │   └── FirebaseConfig.java        | 설정 정의
│       ├── event/
│       │   ├── CreatedAlertEvent.java     | 위험 알림 이벤트 처리
│       │   ├── CreatedEventListener.java  | 이벤트 생성에 대한 전반적 처리
│       │   ├── CreatedMapEvent.java       | 맵 생성 알림 이벤트 처리
│       │   ├── NotificationController.java | 토큰에 의한 권한부여 컨트롤러
│       │   ├── NotificationService.java   | 권한부여 서비스 레이어
│       │   ├── TokenMapper.java           | 권한부여 맵퍼 인터페이스
│       │   └── TokenVO.java               | DB 매핑 모델
│       ├── ...
│       └── common/
├── frontend/
│   ├── nginx/
│   ├── public/
│   │   ├── manifest.json                  | 모바일 앱 설정
│   │   ├── firebase-messaging-sw.js       | 메시지 처리 서비스워커
│   ├──...
│   └── src/
│       ├── firebase/
│       │   └── index.js                   | 파이어베이스 환경 설정
│       ├── ...
│       ├── utils/
│       │   └── firebase.js                | 인증, 메시징 처리 유틸
│       ├── ...
│       ├── App.css
│       ├── App.jsx                        | 서비스워커 호출 및 메시징 처리
│       ├── index.css
│       └── main.jsx
└── robot/
```

DB 덤프 파일

- 마이그레이션 코드

```
drop DATABASE areuhot;

CREATE DATABASE IF NOT EXISTS areuhot CHARACTER SET utf8mb4 COL
LATE utf8mb4_unicode_ci;
USE areuhot;

-- 창고
CREATE TABLE warehouse (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    location VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL
);

-- 로봇
CREATE TABLE robot (
    id INT PRIMARY KEY AUTO_INCREMENT,
    warehouse_id INT NOT NULL,
    name VARCHAR(100),
    status ENUM('ACTIVE', 'INACTIVE') NOT NULL,
    created_at DATETIME NOT NULL,
    FOREIGN KEY (warehouse_id) REFERENCES warehouse(id)
);

-- 로봇 위치 로그
CREATE TABLE robot_log (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    robot_id INT NOT NULL,
    x DOUBLE NOT NULL,
    y DOUBLE NOT NULL,
    direction FLOAT NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    FOREIGN KEY (robot_id) REFERENCES robot(id)
);
CREATE INDEX idx_robot_time ON robot_log(robot_id, created_at);
```

```sql
-- 맵 데이터
CREATE TABLE map (
    id INT PRIMARY KEY AUTO_INCREMENT,
    warehouse_id INT NOT NULL,
    version VARCHAR(50) NOT NULL, -- 날짜 + 시퀀스
    is_active BOOLEAN DEFAULT FALSE NOT NULL,
    type VARCHAR(20) DEFAULT 'processed' NOT NULL,
    file_path VARCHAR(2048) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    FOREIGN KEY (warehouse_id) REFERENCES warehouse(id)
);

-- 랙
CREATE TABLE rack (
    id INT PRIMARY KEY AUTO_INCREMENT,
    warehouse_id INT NOT NULL,
    map_id INT NOT NULL,
    x1 DOUBLE NOT NULL,
    y1 DOUBLE NOT NULL,
    x2 DOUBLE NOT NULL,
    y2 DOUBLE NOT NULL,
    x3 DOUBLE NOT NULL,
    y3 DOUBLE NOT NULL,
    x4 DOUBLE NOT NULL,
    y4 DOUBLE NOT NULL,
    center_x DOUBLE NOT NULL,
    center_y DOUBLE NOT NULL,
    FOREIGN KEY (warehouse_id) REFERENCES warehouse(id),
    FOREIGN KEY (map_id) REFERENCES map(id)
);

-- 촬영 스팟
CREATE TABLE spot (
    id INT PRIMARY KEY AUTO_INCREMENT,
    uuid BINARY(16) NOT NULL UNIQUE,
    rack_id INT NOT NULL,
    x DOUBLE NOT NULL,
```

```
    y DOUBLE NOT NULL,
    direction FLOAT NOT NULL,
    FOREIGN KEY (rack_id) REFERENCES rack(id)
);

-- 관리자
CREATE TABLE user (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE COMMENT '로그인 아이디',
    name VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL,
    role VARCHAR(20) DEFAULT 'ADMIN' NOT NULL,
    is_active BOOLEAN DEFAULT TRUE NOT NULL
);

-- 위험 리포트
CREATE TABLE alert (
    id INT PRIMARY KEY AUTO_INCREMENT,
    robot_id INT NOT NULL,
    rack_id INT NOT NULL,
    warehouse_id INT NOT NULL,
    spot_id INT NOT NULL,
    temperature DOUBLE NOT NULL,
    image_thermal_url VARCHAR(2048) NOT NULL,
    image_normal_url VARCHAR(2048) NOT NULL,
    status ENUM('UNCHECKED', 'DONE') DEFAULT 'UNCHECKED' NOT NUL
L,
    is_danger BOOLEAN NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CU
RRENT_TIMESTAMP,
    FOREIGN KEY (robot_id) REFERENCES robot(id),
    FOREIGN KEY (rack_id) REFERENCES rack(id),
    FOREIGN KEY (warehouse_id) REFERENCES warehouse(id),
    FOREIGN KEY (spot_id) REFERENCES spot(id)
);

CREATE INDEX idx_alert_warehouse_created_status
```

```
ON alert (warehouse_id, created_at DESC, status);

-- 리포트 처리 내역
CREATE TABLE alert_processing (
    id INT PRIMARY KEY AUTO_INCREMENT,
    alert_id INT NOT NULL,
    user_id INT NOT NULL COMMENT '처리한 관리자 계정 ID',
    rack_id INT NOT NULL,
    handled_at DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    handler_name VARCHAR(100) NOT NULL COMMENT '실제 경고를 처리한
작업자 이름',
    item_type VARCHAR(100),
    comment TEXT,
    FOREIGN KEY (alert_id) REFERENCES alert(id),
    FOREIGN KEY (user_id) REFERENCES user(id),
    FOREIGN KEY (rack_id) REFERENCES rack(id)
);

CREATE TABLE robot_command (
        id INT AUTO_INCREMENT PRIMARY KEY,
      created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
      robot_id INT NOT NULL,
      warehouse_id INT NOT NULL,
      spot_id INT NOT NULL,
      command_id BINARY(16) NOT NULL,
      timeout_sec INT NOT NULL,
      FOREIGN KEY (robot_id) REFERENCES robot(id),
    FOREIGN KEY (spot_id) REFERENCES spot(id),
    FOREIGN KEY (warehouse_id) REFERENCES warehouse(id)
);

CREATE TABLE notification_token (
    id INT AUTO_INCREMENT NOT NULL,
    token VARCHAR(255) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY uq_token (token)
);
```

- Mock data

```
-- 창고 데이터
INSERT INTO warehouse (name, location) VALUES
('Seoul Main Warehouse', '1234 Seoul Street'),
('Busan Sub Warehouse', '5678 Busan Avenue');

-- 로봇 데이터
INSERT INTO robot (warehouse_id, name, status, created_at) VALUES
(1, 'AlphaBot', 'ACTIVE', '2025-07-20 09:00:00'),
(2, 'BetaBot', 'ACTIVE', '2025-07-20 10:00:00');

-- 로봇 위치 로그 데이터
INSERT INTO robot_log (robot_id, x, y, direction, created_at) VALUES
(1, 20.13, 38.25, 174.72, '2025-07-23 08:00:00'),
(1, 48.98, 10.68, 80.48, '2025-07-23 08:05:00'),
(1, 10.28, 19.48, 191.96, '2025-07-23 08:10:00'),
(1, 35.25, 25.28, 201.22, '2025-07-23 08:15:00'),
(1, 25.47, 43.64, 284.74, '2025-07-23 08:20:00');

-- 맵 데이터
INSERT INTO map (warehouse_id, version, is_active, type, file_path) VALUE
S
(1, 'v_1_2025_08_14_150001', 1, 'raw', 'maps/1_map_20250814_150001.pgm'),
(2, 'v_2_2025_08_14_150002', 1, 'raw', 'maps/2_map_20250814_150002.pg
m');

-- 랙 데이터
INSERT INTO rack (warehouse_id, map_id, x1, y1, x2, y2, x3, y3, x4, y4, cen
ter_x, center_y) VALUES
(1, 1, 83, 65, 102, 65, 102, 84, 83, 84, 92, 74),
(1, 1, 41, 64, 55, 64, 55, 81, 41, 81, 48, 72),
(2, 2, 61, 56, 78, 56, 78, 69, 61, 69, 69, 62);

-- 촬영 스팟 데이터 (uuid 추가)
INSERT INTO spot (uuid, rack_id, x, y, direction) VALUES
(UUID_TO_BIN(UUID('ed6f7bf9-99c1-4682-ad7e-104513ce9bc9')), 1, 15, 10,
90.00),
```

```
(UUID_TO_BIN(UUID('f6088078-e11c-4d2c-87d3-3f06d530ea54')), 2, 35,
30, 270.00);

-- 관리자 데이터
INSERT INTO user (username, name, password, role, is_active) VALUES
('admin1', '관리자1', '$10$bg2ph5D9im.CEcsu1p5hsuKLJ5w0M3JLVP9FjOe8
qRtF5JMwjB912', 'ADMIN', true),
('admin2', '관리자2', '$10$bg2ph5D9im.CEcsu1p5hsuKLJ5w0M3JLVP9FjOe
8qRtF5JMwjB912', 'ADMIN', false),
('ssafy', '김싸피', '$10$6/tY4NJ0/C8luoUnmb/qsOzCYAt51MLDRqWeHPPNc
aKmPOmWRrSea', 'ADMIN', true);

-- 위험 리포트 데이터 (x, y, direction 컬럼 제거 및 spot_id 추가)
INSERT INTO alert (
  robot_id, rack_id, warehouse_id, spot_id, temperature,
  image_thermal_url, image_normal_url,
  status, is_danger, created_at, updated_at
) VALUES
(1, 1, 1, 1, 78.4, 'photos/thermal_20250818_084738.jpg', 'photos/camera_20
250818_084738.jpg', 'UNCHECKED', false, '2025-07-23 09:00:00', '2025-0
7-23 09:00:00'),
(1, 1, 1, 1, 79.1, 'photos/thermal_20250818_085002.jpg', 'photos/camera_20
250818_085002.jpg', 'UNCHECKED', false, '2025-07-23 10:00:00', '2025-0
7-23 10:00:00'),
(2, 2, 2, 2, 85.6, 'photos/thermal_20250818_085103.jpg', 'photos/camera_2
0250818_085103.jpg', 'DONE', true, '2025-07-23 11:00:00', '2025-07-23 11:
30:00');

-- 리포트 처리 내역 데이터
INSERT INTO alert_processing (
  alert_id, user_id, rack_id, handled_at, handler_name, item_type, comment
) VALUES
(1, 1, 1, '2025-07-23 10:10:00', '김작업자', 'RACK_OVERHEAT', '육안 이상 없음.
경고 해제.'),
(3, 2, 2, '2025-07-23 11:40:00', '박작업자', 'FIRE_DETECTED', '화재 징후 확인.
담당 부서 연락 완료.');
```

시연 시나리오

1. 1차 자율 주행으로 지정된 촬영 위치로 이동
2. 해당 위치에서 1,2,3층을 차례로 랙을 촬영
3. 뜨거운 물체 탐지로 인한 알림 발생
4. 2차 자율 주행으로 지정된 촬영 위치로 이동
5. 해당 위치에서 1,2,3층을 차례로 랙을 촬영
6. 웹에서 알림 확인 및 여러 기능 확인
7. 웹앱에서 알림 생성 확인