포팅 메뉴얼

1. 사용 도구

- 이슈 관리: 노션, Jira
- 형상 관리: Git
- 커뮤니케이션: MatterMost, kakao talk
- 디자인: Figma
- CI/CD: Jenkins

2. 개발 도구

Frontend

- 프레임워크 : React
- 라이브러리 : React-Redux, React Router

Backend

- 프레임워크 : Spring
- 라이브러리 : RabbitMQ, JPA, Spring Security, MySQL Driver, Lombok

Embedded

- 프레임워크 : PyQT, ROS
- 라이브러리 : aiohttp, opencv, paho-mqtt, pika, requests, adafruit, rplidar_ros
- 언어: Python, C++
- 외부 API : SK License Plate Recognition(lpr) openAPI
  https://skopenapi.readme.io/reference/lpr-1
- 라즈베리파이 세팅

▼ Mosquitto 세팅

- Mosquitto 설치

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install mosquitto mosquitto-clients
```

- Mosquitto 환경 세팅

```
# /etc/mosquitto/conf.d 아래에 external_access.conf 명의
sudo nano /etc/mosquitto/conf.d/external_access.conf

# 다음과 같이 작성합니다.
# 외부 접근을 허용하는 설정
listener 1883
allow_anonymous true
```

▼ 라즈베리파이 PyQT6 설치

- xcb 관련 라이브러리 설치

```
sudo apt-get install libxcb-xinerama0 libxcb-cursor0
```

▼ 라즈베리파이 requirements.txt

```
# requirements.txt

aiohappyeyeballs==2.3.5
aiohttp==3.10.3
aiosignal==1.3.1
attrs==24.2.0
certifi==2024.7.4
charset-normalizer==3.3.2
frozenlist==1.4.1
idna==3.7
multidict==6.0.5
numpy==2.0.1
opencv-python==4.10.0.84
```

```
paho-mqtt==2.1.0
pika==1.3.2
PyQt6==6.7.1
PyQt6-Qt6==6.7.2
PyQt6_sip==13.8.0
qasync==0.27.1
requests==2.32.3
urllib3==2.2.2
yarl==1.9.4
```

▼ 라즈베리파이 가상 환경 설정

1. 가상환경 생성 및 실행

```
python -m venv venv

source venv/bin/activate   # Windows의 경우 `venv\Scripts
```

2. 필요한 패키지 설치

```
pip install -r requirements.txt
```

- Jetson Orin 세팅

▼ ROS 설치

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubu
$ sudo apt install curl
$ curl -s https://raw.githubusercontent.com/ros/rosdist
$ sudo apt update
$ sudo apt install ros-noetic-desktop-full
$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
$ echo "source /opt/ros/noetic/setup.zsh" >> ~/.zshrc
$ source ~/.zshrc
$ roscore
```

▼ a2m12 rplidar LiDAR 실행

```
$ roslaunch rplidar_ros view_rplidar_a2m12.launch
```

▼ LiDAR 웹소켓 실행

```
$ roslaunch rosbridge_server rosbridge_websocket.launch
```

▼ Jetson Orin requiremetns.txt

```
# requirements.txt

actionlib==1.14.0
Adafruit-Blinka==8.46.0
adafruit-circuitpython-busdevice==5.2.9
adafruit-circuitpython-connectionmanager==3.1.1
adafruit-circuitpython-motor==3.4.13
adafruit-circuitpython-pca9685==3.4.15
adafruit-circuitpython-register==1.9.18
adafruit-circuitpython-requests==4.1.4
adafruit-circuitpython-servokit==1.3.17
adafruit-circuitpython-typing==1.10.3
Adafruit-GPIO==1.0.3
Adafruit-PCA9685==1.0.1
Adafruit-PlatformDetect==3.72.1
Adafruit-PureIO==1.1.11
alabaster==0.7.8
angles==1.9.13
apt-clone==0.2.1
apturl==0.5.2
attrs==19.3.0
autobahn==17.10.1
Automat==0.8.0
Babel==2.6.0
base-local-planner==1.17.3
bcrypt==3.1.7
binho-host-adapter==0.1.6
blinker==1.4
bondpy==1.8.6
breezy==3.0.2
```

```
Brlapi==0.7.0
camera-calibration==1.17.0
camera-calibration-parsers==1.12.0
catkin==0.8.10
catkin-pkg==1.0.0
catkin-pkg-modules==1.0.0
cbor==1.0.0
certifi==2019.11.28
chardet==3.0.4
Click==7.0
colorama==0.4.3
configobj==5.0.6
constantly==15.1.0
controller-manager==0.20.0
controller-manager-msgs==0.20.0
cryptography==2.8
cupshelpers==1.0
cv-bridge==1.16.2
cycler==0.10.0
Cython==0.29.14
dbus-python==1.2.16
decorator==4.4.2
defer==1.0.6
defusedxml==0.6.0
Deprecated==1.2.7
diagnostic-analysis==1.11.0
diagnostic-common-diagnostics==1.11.0
diagnostic-updater==1.11.0
distro==1.4.0
distro-info==0.23+ubuntu1.1
docutils==0.16
dulwich==0.19.15
duplicity==0.8.12.0
dynamic-reconfigure==1.7.3
empy==3.3.2
entrypoints==0.3
evdev==1.7.1
fasteners==0.14.1
```

```
fastimport==0.9.8
future==0.18.2
gazebo-plugins==2.9.2
gazebo-ros==2.9.2
gencpp==0.7.0
geneus==3.0.0
genlisp==0.4.18
genmsg==0.6.0
gennodejs==2.0.2
genpy==0.6.15
gpg==1.13.1
httplib2==0.14.0
hyperlink==19.0.0
idna==2.8
image-geometry==1.16.2
imagesize==1.2.0
incremental==16.10.1
interactive-markers==1.12.0
jetson-stats==4.2.8
Jetson.GPIO==2.1.6
Jinja2==2.10.1
joint-state-publisher==1.15.1
joint-state-publisher-gui==1.15.1
keyring==18.0.1
kiwisolver==1.0.1
language-selector==0.1
laser-geometry==1.6.7
launchpadlib==1.10.13
lazr.restfulclient==0.14.2
lazr.uri==1.0.3
lockfile==0.12.2
louis==3.12.0
lz4==3.0.2+dfsg
macaroonbakery==1.3.1
Mako==1.1.0
MarkupSafe==1.1.0
matplotlib==3.1.2
message-filters==1.16.0
```

```
monotonic==1.5
mpi4py==3.0.3
netifaces==0.10.4
nose==1.3.7
numpy==1.19.5
oauthlib==3.1.0
olefile==0.46
onboard==1.4.1
packaging==20.3
paho-mqtt==2.1.0
PAM==0.4.2
pandas==0.25.3
paramiko==2.6.0
pexpect==4.6.0
Pillow==7.0.0
protobuf==3.6.1
psutil==5.5.1
py-ubjson==0.14.0
pyasn1==0.4.2
pyasn1-modules==0.2.1
pycairo==1.16.2
pycrypto==2.6.1
pycryptodomex==3.6.1
pycups==1.9.73
pydot==1.4.1
pyftdi==0.55.4
PyGithub==1.43.7
Pygments==2.3.1
PyGObject==3.36.0
PyHamcrest==1.9.0
PyICU==2.4.2
PyJWT==1.7.1
pymacaroons==0.13.0
PyNaCl==1.3.0
pynput==1.7.7
PyOpenGL==3.1.0
pyOpenSSL==19.0.0
pyparsing==2.4.6
```

```
pypng==0.0.20
PyQRCode==1.2.1
PyQt5==5.14.1
pyRFC3339==1.1
pyserial==3.5
python-apt==2.0.1+ubuntu0.20.4.1
python-dateutil==2.7.3
python-dbusmock==0.19
python-debian==0.1.36+ubuntu1.1
python-gitlab==2.0.1
python-gnupg==0.4.5
python-qt-binding==0.4.4
python-snappy==0.5.3
python-xlib==0.33
PyTrie==0.2
pytz==2019.3
pyusb==1.2.1
pyxdg==0.26
PyYAML==5.3.1
qt-dotgraph==0.4.2
qt-gui==0.4.2
qt-gui-cpp==0.4.2
qt-gui-py-common==0.4.2
requests==2.22.0
requests-unixsocket==0.2.0
resource-retriever==1.12.7
roman==2.0.0
rosapi==0.11.17
rosbag==1.16.0
rosboost-cfg==1.15.8
rosbridge-library==0.11.17
rosbridge-server==0.11.17
rosclean==1.15.8
roscreate==1.15.8
rosdep==0.24.0
rosdep-modules==0.24.0
rosdistro==0.9.1
rosdistro-modules==0.9.1
```

```
rosgraph==1.16.0
rosinstall==0.7.8
rosinstall-generator==0.1.23
roslaunch==1.16.0
roslib==1.15.8
roslint==0.12.0
roslz4==1.16.0
rosmake==1.15.8
rosmaster==1.16.0
rosmsg==1.16.0
rosnode==1.16.0
rosparam==1.16.0
rospkg==1.5.1
rospkg-modules==1.5.1
rospy==1.16.0
rosservice==1.16.0
rostest==1.16.0
rostopic==1.16.0
rosunit==1.15.8
roswtf==1.16.0
rplidar-roboticia==0.9.5
rqt-action==0.4.9
rqt-bag==0.5.1
rqt-bag-plugins==0.5.1
rqt-console==0.4.12
rqt-dep==0.4.12
rqt-graph==0.4.14
rqt-gui==0.5.3
rqt-gui-py==0.5.3
rqt-image-view==0.4.17
rqt-launch==0.4.9
rqt-logger-level==0.4.12
rqt-moveit==0.5.11
rqt-msg==0.4.10
rqt-nav-view==0.5.7
rqt-plot==0.4.13
rqt-pose-view==0.5.11
rqt-publisher==0.4.10
```

```
rqt-py-common==0.5.3
rqt-py-console==0.4.10
rqt-reconfigure==0.5.5
rqt-robot-dashboard==0.5.8
rqt-robot-monitor==0.5.15
rqt-robot-steering==0.5.12
rqt-runtime-monitor==0.5.10
rqt-rviz==0.7.0
rqt-service-caller==0.4.10
rqt-shell==0.4.11
rqt-srv==0.4.9
rqt-tf-tree==0.6.4
rqt-top==0.4.10
rqt-topic==0.4.13
rqt-web==0.4.10
rviz==1.14.25
scipy==1.3.3
SecretStorage==2.3.1
sensor-msgs==1.13.1
service-identity==18.1.0
simplejson==3.16.0
sip==4.19.21
six==1.14.0
smach==2.5.2
smach-ros==2.5.2
smbus2==0.4.3
smclib==1.8.6
Sphinx==1.8.5
spidev==3.6
systemd-python==234
sysv-ipc==1.1.0
tf==1.13.2
tf-conversions==1.13.2
tf2-geometry-msgs==0.7.7
tf2-kdl==0.7.7
tf2-py==0.7.7
tf2-ros==0.7.7
topic-tools==1.16.0
```

```
tornado==5.1.1
Twisted==18.9.0
txaio==2.10.0
typing-extensions==4.12.2
u-msgpack-python==2.1
ubuntu-advantage-tools==8001
ubuntu-drivers-common==0.0.0
urllib3==1.25.8
urwid==2.0.1
vcstools==0.1.42
wadllib==1.3.3
websocket-client==1.8.0
wrapt==1.11.2
wsaccel==0.6.2
wstool==0.1.18
xacro==1.14.18
xkit==0.0.0
zope.interface==4.7.1
```

Aiot

- 언어 : Python
- 라이브러리 : Gymnasium, stable-baselines3, highway-env, CUDA, numpy, matplotlib, pickle

3. 개발 환경

Frontend

TypeScript	`4.9.5`
React	`18.3.1`
CSS	`3`
reduxjs/toolkit	`2.2.7`

VS code	`1.90.2`

Backend

Java	`17`
Spring Boot	`3.3.1`
MySql	`8`
RabbitMQ	`3`
IntelliJ IDEA	`2024.1.4`

Embedded

Python	`3.11.2`
PyQT	`6`
Mosquitto	`2`

Aiot

Python	`3.9.13`
stable-baselines	`2.3.2`
highway-env	`1.8.2`
CUDA	`12.1+cu118`
VS code	`1.90.12`
gymnasium	`0.29.1`
numpy	`1.26.3`

Infra

Docker	`27.0.3`
Nginx	`1.18.0`
Jenkins	**`2.452.3`**
RabbitMQ	`3`

4. 환경 변수

Backend

application.properties

```
spring.profiles.include=local
```

application-local.properties

```
spring.datasource.url=jdbc:mysql://db:3306/mvpdb
spring.datasource.username=root
spring.datasource.password=mvp2442
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect
server.port=8081
```

Embedded

- 키오스크 (config.py)

```
# config/config.py
RABBIT_MQ_HOST = "mvp-project.shop"
RABBIT_Q_NAME = "task_queue"
MQTT_PORT = 1883
MQTT_BROKER_IP = "localhost"
MQTT_TOPIC_SUB = "/parking/complete"
MQTT_TOPIC_PUB = "/parking/command"
MQTT_BROKER_PATH_WIN = "C:\\Program Files\\mosquitto\\mosquit
MQTT_CONF_PATH_WIN = "C:\\Program Files\\mosquitto\\mosquitto
MQTT_BROKER_PATH_RASP = "/usr/sbin/mosquitto"
MQTT_CONF_PATH_RASP = "/etc/mosquitto/mosquitto.conf"
SERVER_URL = "https://mvp-project.shop/api"
API_KEY = ""
API_KEY_LPR = "l7xx846db5f3bc1e48d29b7275a745d501c8"
```

- Orin (config.py)

```
# config/config.py
MQTT_BROKER = "192.168.137.63" #라즈베리파이 브로커 IP
MQTT_PORT = 1883
MQTT_TOPIC_PUB = "/parking/complete"
MQTT_TOPIC_SUB = "/parking/command"
SERIAL_NUMBER = 1001
```

5. CI/CD

AWS

- 포트 번호

MySQL	3306
Jenkins	8080
Backend	8081
Nginx	80
React	3000
RabbitMQ	5672

Docker

도커 설치

패키지 업데이트

```
sudo apt-get update
```

필요한 패키지 설치

```
sudo apt-get install apt-transport-https ca-certificates curl
```

gpg키 추가

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sud
```

도커 저장소 설정

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/shar
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list
```

도커 설치

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

도커 실행

```
sudo service docker start
```

도커 컴포즈 설치

```
sudo curl -L https://github.com/docker/compose/releases/downl
sudo chmod +x /usr/local/bin/docker-compose
sudo usermod -aG docker $USER
```

```
docker-compose --version
```

Jenkins

젠킨스 설치

볼륨 디렉토리 생성

```
cd /home/ubuntu && mkdir jenkins-data
```

외부에서 접속할 포트 오픈 및 상태 확인

```
sudo ufw allow 8080/tcp
sudo ufw reload
sudo ufw status
```

젠킨스 컨테이너 생성 및 구동

```
sudo docker run -d -p 8080:8080 -v /var/run/docker.sock:/var/
```

로그 확인 및 초기 패스워드 확인

```
sudo docker logs jenkins
```

젠킨스 계정 설정

기억해둔 초기 패스워드를 이용해서 계정 설정을 진행합니다.

초기 비밀번호 확인

```
sudo docker logs jenkins
```

젠킨스 접속 및 계정 설정

```
http://주소:8080
```

젠킨스 플러그인 설치

제안 플러그인을 모두 설치합니다.

추가 플러그인을 설치합니다.

```
- Docker
- Docker pipeline
- Docker API
- Gitlab
- Generic Webhook Trigger Plugin
- SSH Agent
```

젠킨스 credential 설정

[젠킨스 관리 - Credential]

application-local 등록

```
종류 : Secret file
파일 : application-local.properties
ID : application-local
```

docker-compose 등록

docker-compose.yml

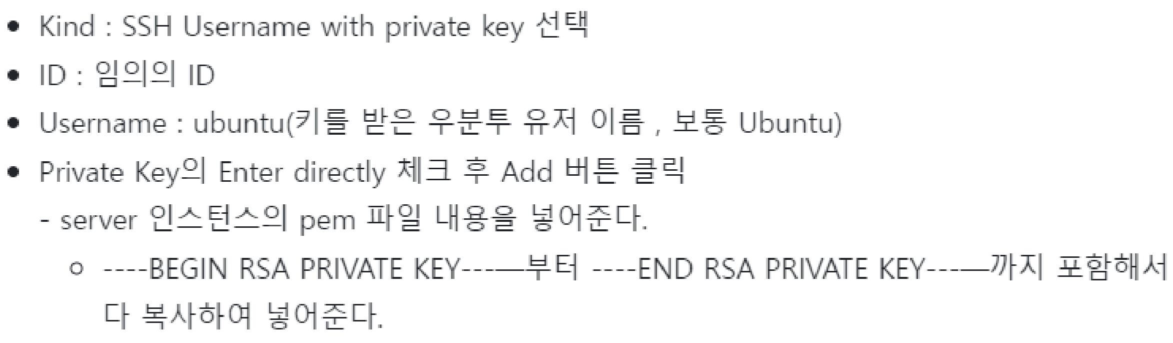
```
종류 : Secret file
파일 : docker-compose.yml
ID : docker-compose
```

docker-compose-fe 등록

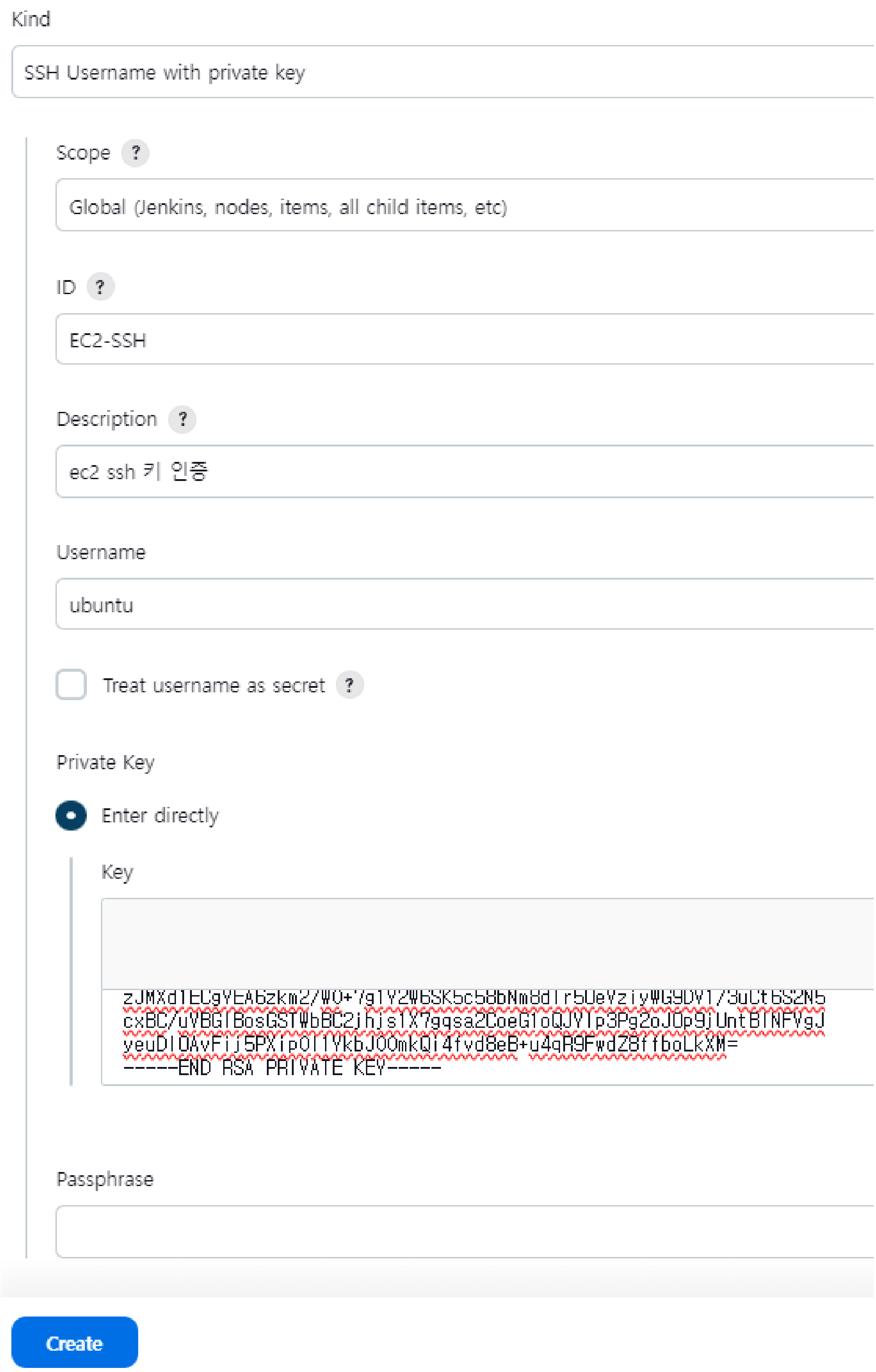
docker-compose-fe.yml

```
종류 : Secret file
파일 : docker-compose-fe.yml
ID : docker-compose-fe
```

ec2_ssh_key 설정



- Kind : SSH Username with private key 선택
- ID : 임의의 ID
- Username : ubuntu(키를 받은 우분투 유저 이름 , 보통 Ubuntu)
- Private Key의 Enter directly 체크 후 Add 버튼 클릭
  - server 인스턴스의 pem 파일 내용을 넣어준다.
  - ----BEGIN RSA PRIVATE KEY---—부터 ----END RSA PRIVATE KEY---—까지 포함해서 다 복사하여 넣어준다.



Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

EC2-SSH

Description ?

ec2 ssh 키 인증

Username

ubuntu

□ Treat username as secret ?

Private Key

● Enter directly

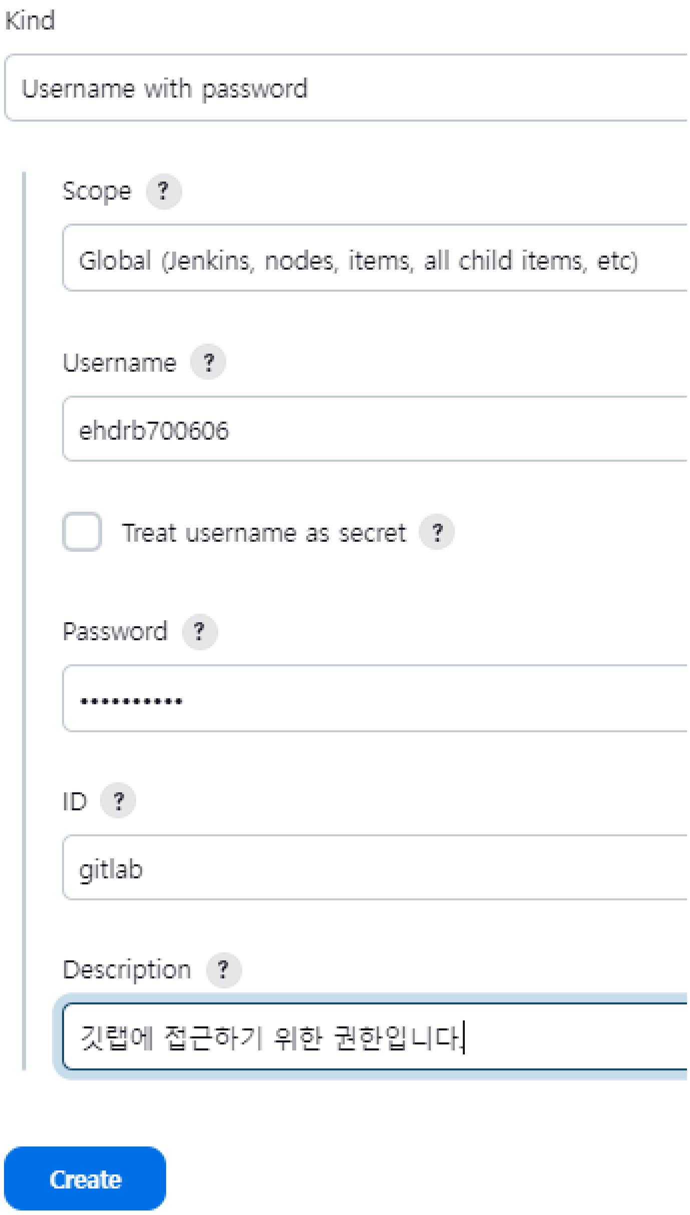
Key

```
zJMXd1ECgYEA6zkm2/₩0+'7g1Y2₩6SK5c58bNm8dIr5UeVziy₩G9DV17/3uCt6S2N5
cxBC/uVBGIBosGST₩bBC2jhjs1X7gqsa2CoeG1oQJYIp3Pg2oJ0p9jUntBINFVgJ
yeuDlOAvFij5PXipOl1YkbJ0OmkQi4fvd8eB+u4qR9FwdZ8ffboLkXM=
-----END RSA PRIVATE KEY-----
```

Passphrase

Create

gitlab_token 등록



Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

ehdrb700606

□ Treat username as secret ?

Password ?

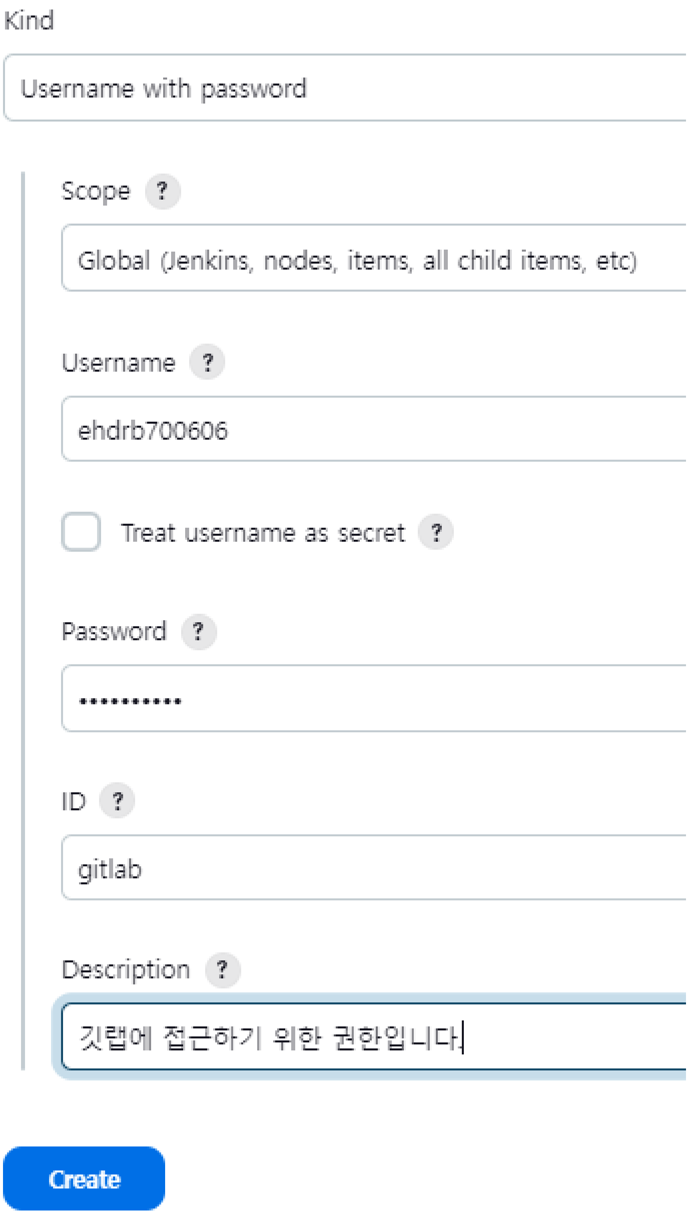
••••••••••

ID ?

gitlab

Description ?

깃랩에 접근하기 위한 권한입니다.



Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

ehdrb700606

□ Treat username as secret ?

Password ?

•••••••••••

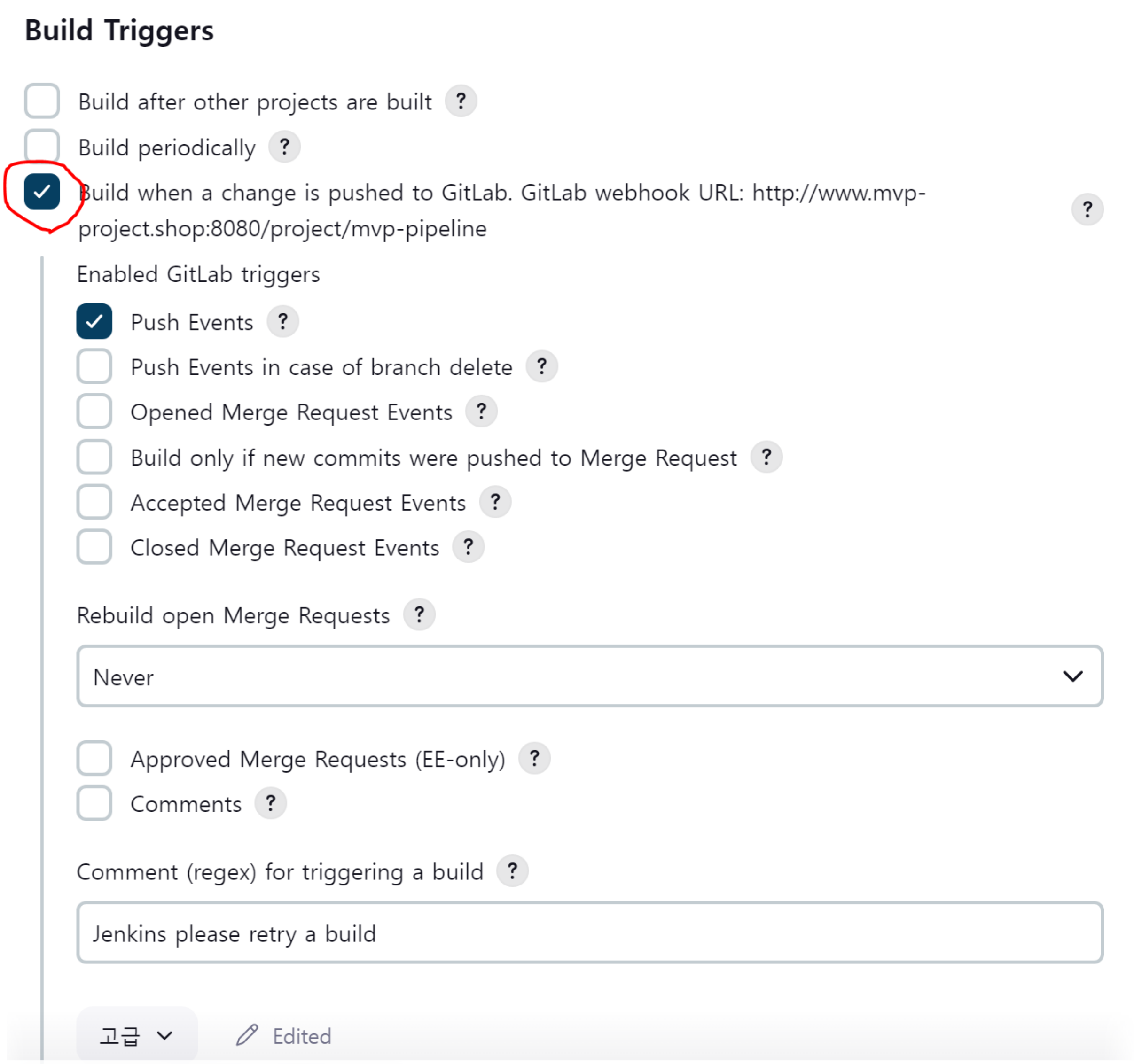
ID ?

gitlab

Description ?

깃랩에 접근하기 위한 권한입니다.

웹훅 설정



Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☑ Build when a change is pushed to GitLab. GitLab webhook URL: http://www.mvp-project.shop:8080/project/mvp-pipeline ?

Enabled GitLab triggers

☑ Push Events ?

☐ Push Events in case of branch delete ?

☐ Opened Merge Request Events ?

☐ Build only if new commits were pushed to Merge Request ?

☐ Accepted Merge Request Events ?

☐ Closed Merge Request Events ?

Rebuild open Merge Requests ?

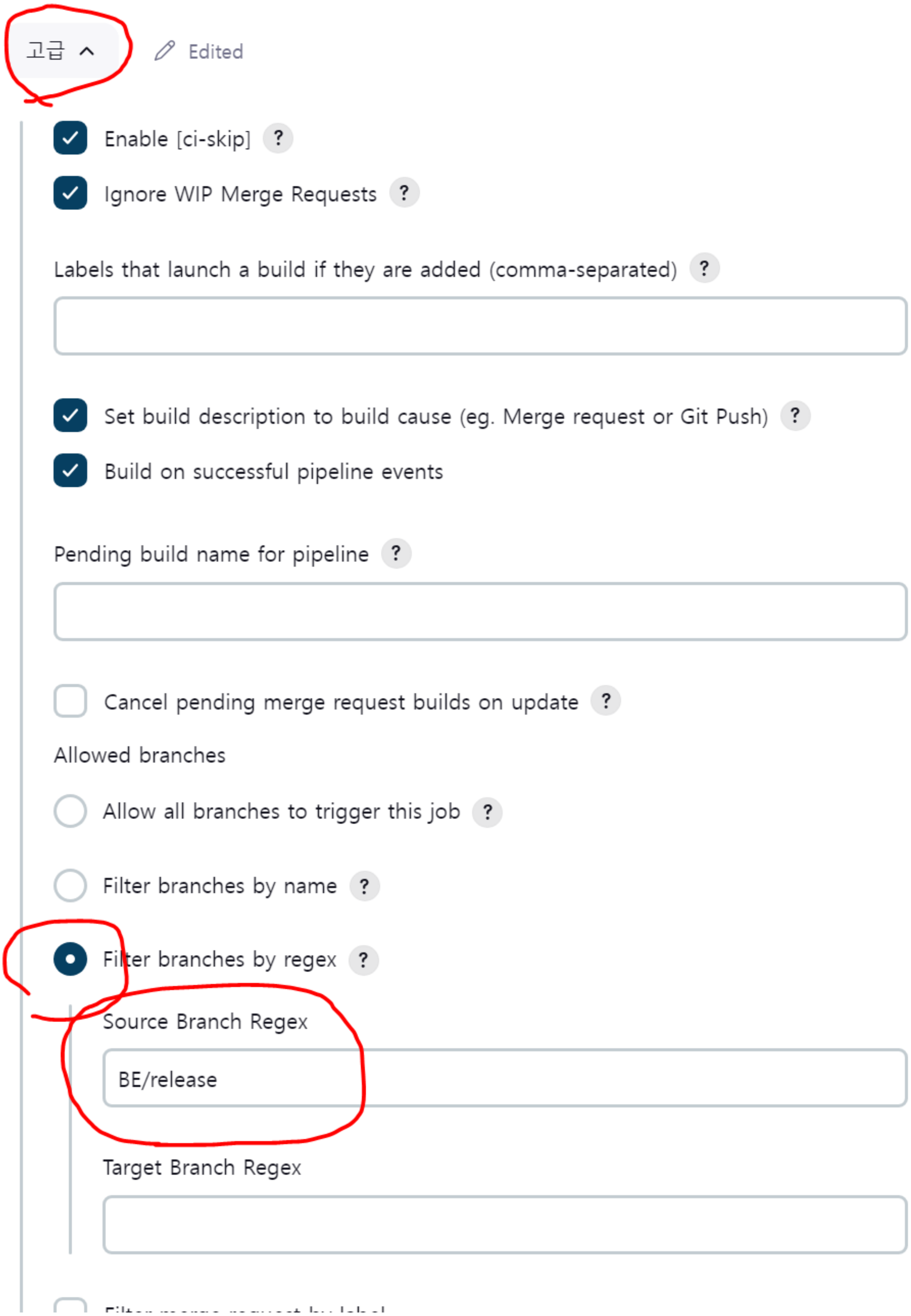
Never

☐ Approved Merge Requests (EE-only) ?

☐ Comments ?

Comment (regex) for triggering a build ?

Jenkins please retry a build



고급 ^ Edited

☑ Enable [ci-skip] ?

☑ Ignore WIP Merge Requests ?

Labels that launch a build if they are added (comma-separated) ?

☑ Set build description to build cause (eg. Merge request or Git Push) ?

☑ Build on successful pipeline events

Pending build name for pipeline ?

□ Cancel pending merge request builds on update ?

Allowed branches

○ Allow all branches to trigger this job ?

○ Filter branches by name ?

● Filter branches by regex ?

Source Branch Regex

BE/release

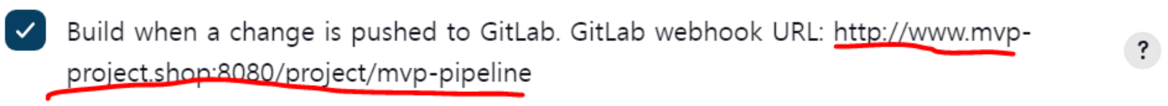
Target Branch Regex

□ Filter merge request by label

다음과 같이 설정 후 시크릿 키를 버튼을 눌러 생성하고 저장해둔다.

GitLab Repository → Settings → WebHooks → add new Webhooks

- URL



☑ Build when a change is pushed to GitLab. GitLab webhook URL: http://www.mvp-project.shop:8080/project/mvp-pipeline

- Secret token 방금 생성한 토큰 입력
- Trigger : Push events - All branches

백엔드 파이프라인과 프론트 파이프라인을 분리해둬서 위와 같은 작업을 똑같이 진행한다.

파이프라인 [백엔드]

```
pipeline {
    agent any

    stages {

        stage('docker, docker-compose install') {
            steps{
                script {
                    sh  """
                        if ! command -v docker > /dev/null; t
                            curl -fsSL https://get.docker.com
                            sh get-docker.sh
                        fi
                    """
                    sh """
                    export PATH=\$PATH:\$HOME/bin
                    mkdir -p \$HOME/bin
                    if ! command -v docker-compose &> /dev/nu
                    then
                        echo "docker-compose could not be fou
                        curl -L "https://github.com/docker/co
                        chmod +x \$HOME/bin/docker-compose
                    else
                        echo "docker-compose is already insta
                    fi
                    """
                }
            }

        }

        stage('docker_stop_remove') {
            steps {
                script {
                    // 실행 중인 컨테이너를 중지
                    sh """
                        if [ \$(docker ps -q -f name=mvp_spri
                            echo "Stopping mvp_spring_app_1 c
                            docker stop mvp_spring_app_1
                        else
                            echo "mvp_spring_app_1 container 
                        fi
                    """
                }
                script {
                    // 중지된 컨테이너를 제거하고 mvp-project 이미지
                    sh """
                        if [ \$(docker ps -a -q -f name=mvp_s
                            echo "Removing mvp_spring_app_1 c
                            docker rm mvp_spring_app_1
                        else
                            echo "mvp_spring_app_1 container 
                        fi

                        if [ \$(docker images -q mvp-project:
                            echo "Removing mvp-project image"
                            docker rmi -f mvp-project:latest
                        else
                            echo "mvp-project image is not fo
                        fi
                    """
                }
            }
        }

        stage('gitlab_clone') {
            steps {
                git branch: 'BE/release', credentialsId: 'git
                script {
                    sh 'git checkout BE/release'

                }
            }
        }

        stage('properties copy'){
            steps{
                withCredentials([file(credentialsId: 'applica
                    script {
                        sh 'pwd'
                        sh 'ls'
                        sh 'chmod +r $properties'
                        sh 'chmod -R 777 BE/mvp/src/main/reso
                        sh 'cp $properties BE/mvp/src/main/re
                        sh 'ls'
                    }
                }

            }
        }

        stage('docker-compose copy'){
            steps{
                withCredentials([file(credentialsId: 'docker-
                    script {
                        sh 'pwd'
                        sh 'ls'
                        sh 'chmod +r $composeFile'
                        sh 'chmod -R 777 BE/mvp'
                        sh 'cp $composeFile BE/mvp/docker-com
                        sh 'ls BE/mvp'
                    }
                }

            }
        }



        stage('build') {
            steps {
                script {
                    dir('BE/mvp') {
                        // Check and remove existing mvp-proj
                        sh """
                            if [ \$(docker images -q mvp-proj
                                echo "Removing existing mvp-p
                                docker rmi mvp-project:latest
                            else
                                echo "mvp-project:latest imag
                            fi
                        """
                        sh "chmod +x ./gradlew"
                        sh "./gradlew clean build"
                    }
                }
            }
        }

        stage('docker_build') {
            steps {
                script {
                    dir('BE/mvp') {
                        sh "docker build -t mvp-project ." //
                    }
                }
            }
        }
        stage('docker_deploy') {
            steps {
                script {
                    dir('BE/mvp') {

                        sh "docker-compose -f docker-compose.

                    }
                }
            }
        }



    }
}
```

파이프라인 [프론트엔드]

```
pipeline {
    agent any

    stages {

        stage('docker, docker-compose install') {
            steps{
                script {
                    sh  """
                        if ! command -v docker > /dev/null; t
                            curl -fsSL https://get.docker.com
                            sh get-docker.sh
                        fi
                    """
                    sh """
                    export PATH=\$PATH:\$HOME/bin
                    mkdir -p \$HOME/bin
                    if ! command -v docker-compose &> /dev/nu
                    then
                        echo "docker-compose could not be fou
                        curl -L "https://github.com/docker/co
                        chmod +x \$HOME/bin/docker-compose
                    else
                        echo "docker-compose is already insta
                    fi
                    """
                }
            }

        }

        stage('docker_stop_remove') {
            steps {
                script {
                    // 실행 중인 컨테이너를 중지
                    sh """
                        if [ \$(docker ps -q -f name=front) ]
                            echo "Stopping front container"
                            docker stop front
                        else
                            echo "mvp_front container is not
                        fi
                    """
                }
                script {
                    // 중지된 컨테이너를 제거하고 mvp-project 이미지
                    sh """
                        if [ \$(docker ps -a -q -f name=front
                            echo "Removing front container"
                            docker rm front
                        else
                            echo "front container is not foun
                        fi

                        if [ \$(docker images -q mvp_frontend
                            echo "Removing mvp_frontend image
                            docker rmi -f mvp_frontend || tru
                        else
                            echo "mvp_frontend image is not f
                        fi
                    """
                }
            }
        }

        stage('gitlab_clone') {
            steps {
                git branch: 'FE/develop', credentialsId: 'git
                script {
                    sh 'git checkout FE/develop'

                }
            }
        }

        stage('docker-compose copy'){
            steps{
                withCredentials([file(credentialsId: 'docker-
                    script {
                        sh 'pwd'
                        sh 'ls'
                        sh 'chmod +r $composeFile'
                        sh 'chmod -R 777 FE/mvp'
                        sh 'cp $composeFile FE/mvp/docker-com
                        sh 'ls FE/mvp'
                    }
                }

            }
        }

        stage('docker_build') {
            steps {
                script {
                    dir('FE/mvp') {
                        sh "docker-compose -f docker-compose.
                    }
                }
            }
        }

    }
}
```

Nginx

NginX 설치

```
sudo apt update
sudo apt install nginx -y
```

nginx 설정 파일 수정

```
sudo vi /etc/nginx/sites-available/default
```

- default

  ```
  server {
      listen 80;
      server_name mvp-project.shop;

      location /api {
          return 301 https://$host:8081$request_uri;
      }

      location / {
          return 301 https://$host$request_uri;
      }
  }

  server {
      listen 443 ssl http2;
  ```

```
    server_name mvp-project.shop;

    ssl_certificate /etc/letsencrypt/live/mvp-project.shop
    ssl_certificate_key /etc/letsencrypt/live/mvp-project.

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forw
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /api {
        proxy_pass http://127.0.0.1:8081;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forw
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

6. 배포 단계

Backend

- Dockerfile

```
FROM openjdk:17
WORKDIR /app
COPY build/libs/*.jar mvp-project.jar
CMD ["java", "-jar", "mvp-project.jar"]
```

Frontend

- Dockerfile

```
FROM node:13.12.0-alpine
COPY package.json package-lock.json ./
RUN npm install
COPY . ./
EXPOSE 3000
```

설치

1. 리포지토리를 클론합니다:

```bash
git clone <repository-url>
cd parking_kiosk
```

2. 가상 환경을 생성하고 활성화합니다:

```bash
python -m venv venv
source venv/bin/activate   # Windows의 경우 `venv\Scripts\activate`
```

3. 필요한 패키지를 설치합니다:

```bash
pip install -r requirements.txt
```

더미데이터 및 프로퍼티 파일 목록

더미데이터.sql

docker-compose.yml

docker-compose-fe.yml

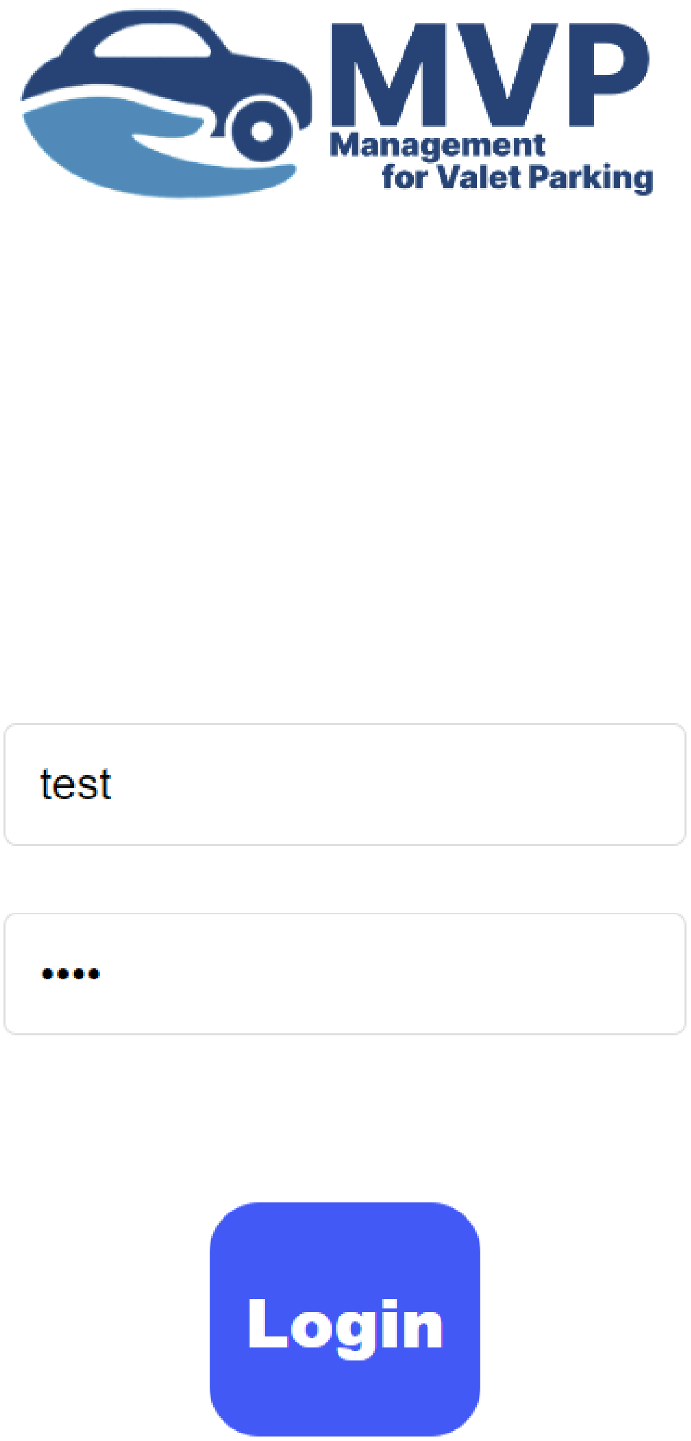
application-local.properties

application.properties

```
spring.profiles.include=local
```

시연 시나리오

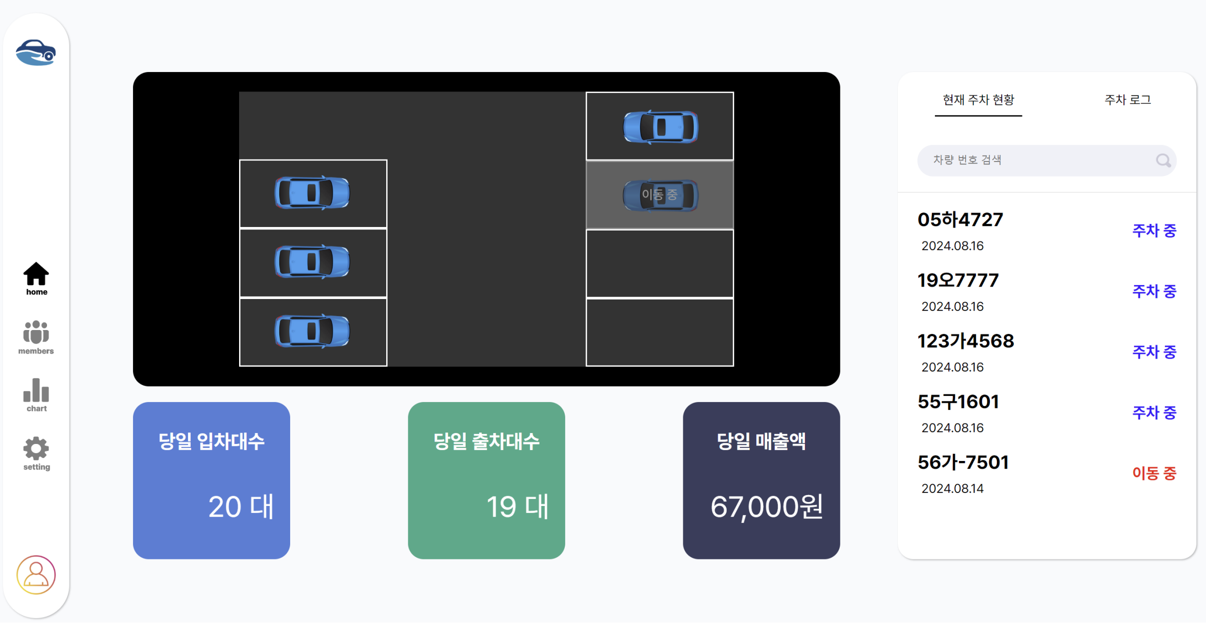
페이지 접속 후 로그인
ID : mvp@naver.com
PW : 1234



test

••••

왼쪽 Nav바의 home/members/chart/setting을 눌러 메인페이지, 회원권 관리 페이지, 통계페이지, 주차장설정 페이지로 갈 수 있습니다.



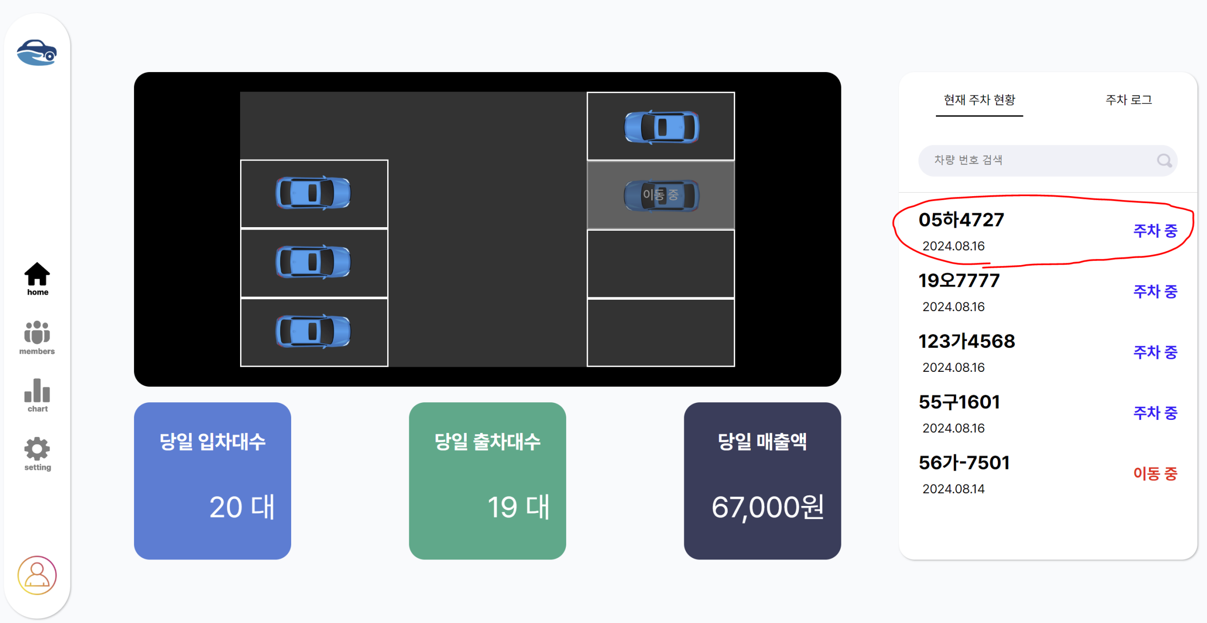
당일 입차대수 20 대

당일 출차대수 19 대

당일 매출액 67,000원

05하4727 2024.08.16	주차 중
19오7777 2024.08.16	주차 중
123가4568 2024.08.16	주차 중
55구1601 2024.08.16	주차 중
56가-7501 2024.08.14	이동 중

메인페이지에서 현재 주차 현황, 주차로그를 눌러 기록을 확인할 수 있고 각 항목을 눌러 자세하게 확인할 수 있습니다.



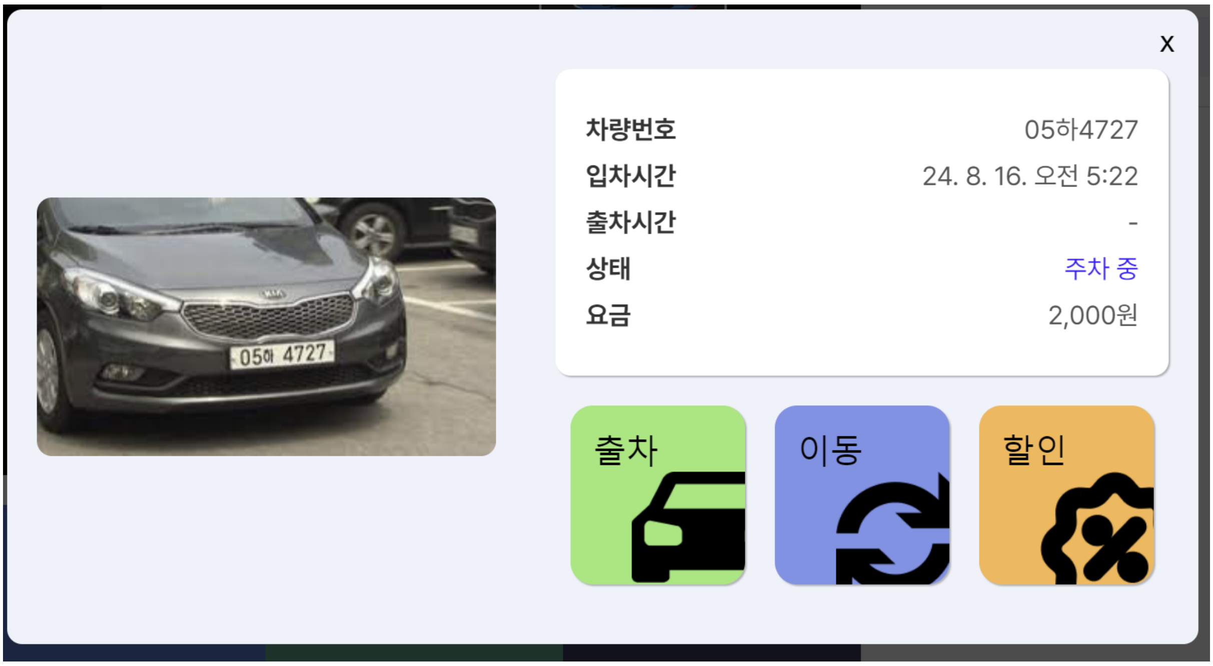
당일 입차대수 20 대

당일 출차대수 19 대

당일 매출액 67,000원

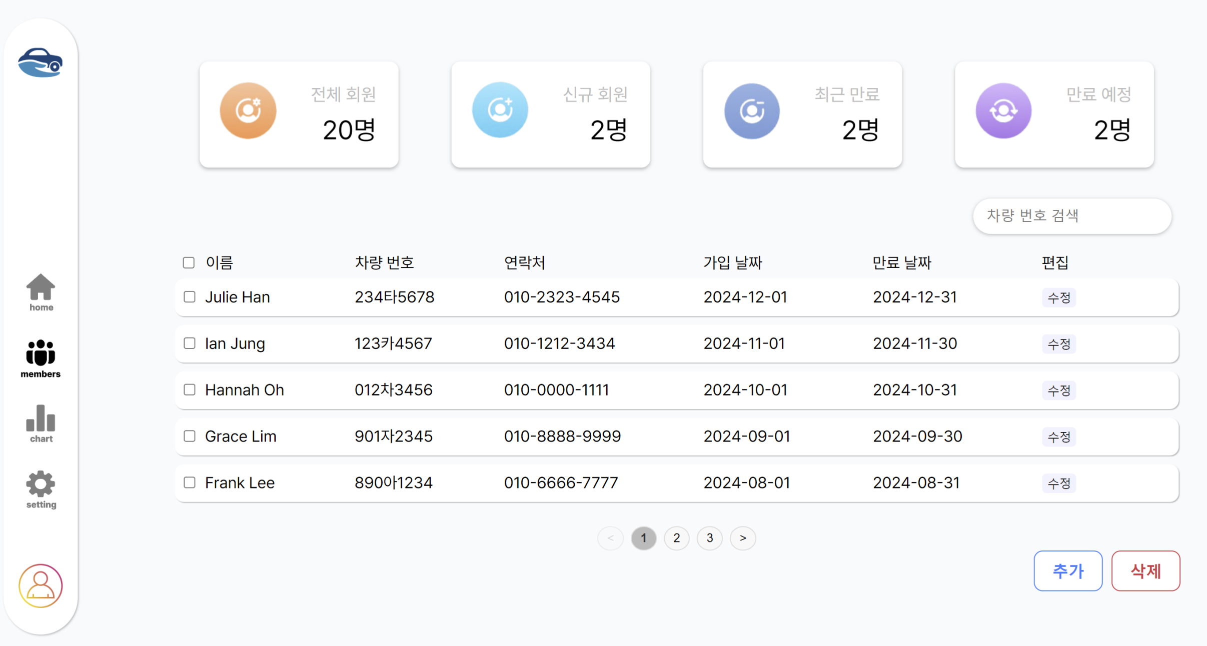
05하4727	2024.08.16	주차 중
19오7777	2024.08.16	주차 중
123가4568	2024.08.16	주차 중
55구1601	2024.08.16	주차 중
56가-7501	2024.08.14	이동 중

상세 보기 모달에서 출차버튼을 누르면 출차 명령이 수행되고,
이동 버튼을 누른 뒤 빈 공간의 주차칸을 선택하면 해당 위치로 차량 위치를 이동하라는 명령을 수행합니다.
할인 버튼을 눌러 요금에 할인을 적용할 수 있습니다.



차량번호	05하4727
입차시간	24. 8. 16. 오전 5:22
출차시간	-
상태	주차 중
요금	2,000원

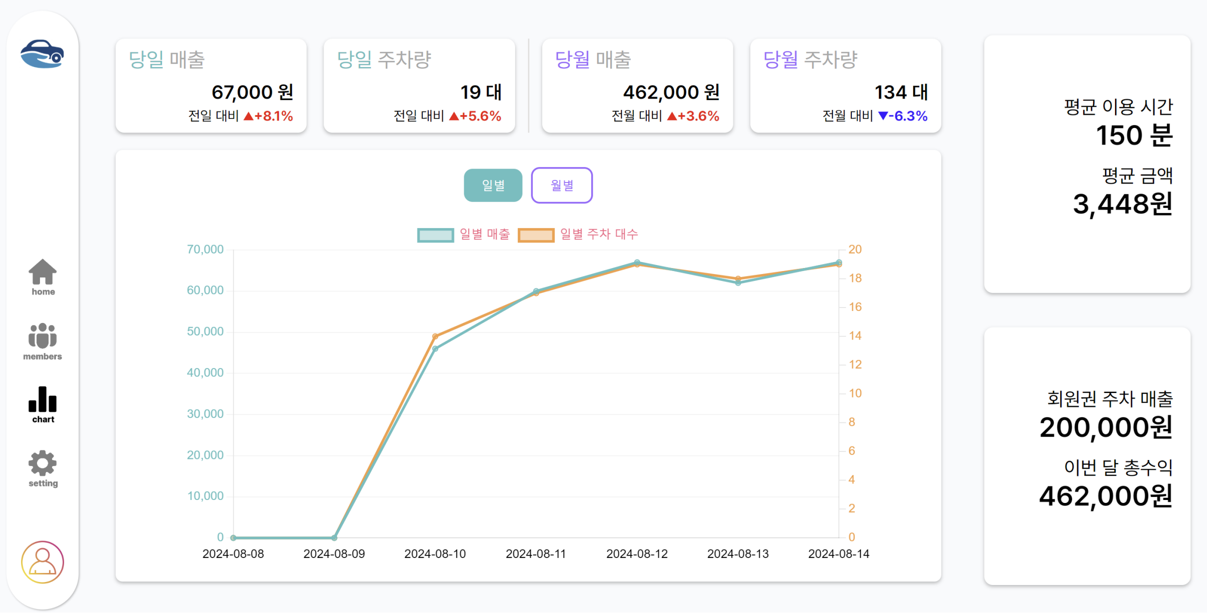
회원 관리 페이지에서 수정을 눌러 회원 정보를 수정할 수 있고 추가/삭제가 가능합니다.



전체 회원	신규 회원	최근 만료	만료 예정
20명	2명	2명	2명

☐ 이름	차량 번호	연락처	가입 날짜	만료 날짜	편집
☐ Julie Han	234타5678	010-2323-4545	2024-12-01	2024-12-31	수정
☐ Ian Jung	123카4567	010-1212-3434	2024-11-01	2024-11-30	수정
☐ Hannah Oh	012차3456	010-0000-1111	2024-10-01	2024-10-31	수정
☐ Grace Lim	901자2345	010-8888-9999	2024-09-01	2024-09-30	수정
☐ Frank Lee	890아1234	010-6666-7777	2024-08-01	2024-08-31	수정

통계 페이지에서 일 별 월 별을 선택해서 각 기간 별 통계를 볼 수 있습니다.



당일 매출
67,000 원
전일 대비 ▲+8.1%

당일 주차량
19 대
전일 대비 ▲+5.6%

당월 매출
462,000 원
전월 대비 ▲+3.6%

당월 주차량
134 대
전월 대비 ▼-6.3%

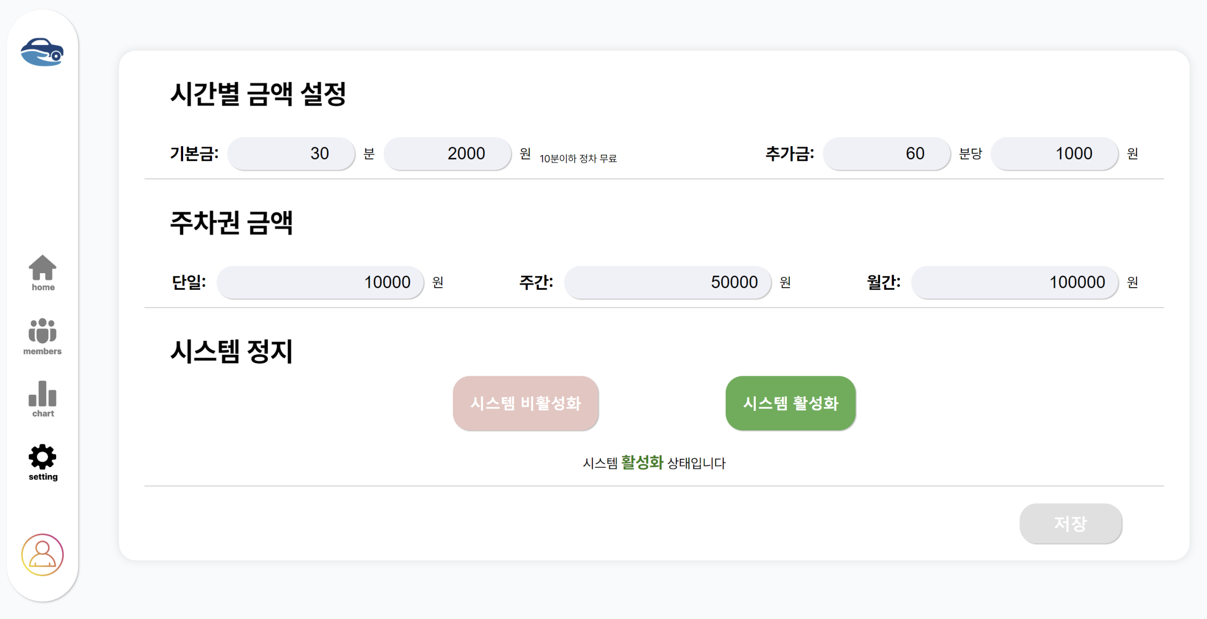
평균 이용 시간
150 분

평균 금액
3,448원

회원권 주차 매출
200,000원

이번 달 총수익
462,000원

설정 페이지에서 주차장 설정을 수정할 수 있습니다.



시간별 금액 설정

기본금: 30 분 2000 원 10분이하 정차 무료

추가금: 60 분당 1000 원

주차권 금액

단일: 10000 원

주간: 50000 원

월간: 100000 원

시스템 정지

시스템 활성화 상태입니다