



## **E201 :**

삼성SW청년아카데미 부울경캠퍼스 7기  
롯데정보통신 기업연계  
자율프로젝트(6주; 2022.10.10 ~ 2022.11.18)

## **포팅 매뉴얼**

담당 컨설턴트 : 김성재  
멘토 : 최은삼 수석, 김광열 선임,  
권도건(팀장), 공지훈, 김동우, 박유주, 장창완, 최지은

## 목차

1. 프로젝트 개요.....	2
2. 프로젝트 기술 스택.....	2
3. 주요 환경 변수.....	4
4. 도커 이미지 빌드 및 실행.....	5
5. 배포 특이사항.....	6
6. 외부 서비스.....	8

### 1. 프로젝트 개요

#### 가. 배경

- 본 프로젝트는 '롯데정보통신'과의 기업 연계 프로젝트로, 쇼핑몰 플랫폼 구축 경험을 통해 쇼핑몰 분야 도메인 지식 함양 및 SI 프로젝트 실무 역량을 향상하고자 함

#### 나. 목표

- 1) 쇼핑몰 플랫폼을 구성하는 주요 요소들의 설계 및 구현
- 2) 사용자별 부여된 권한에 따른 메뉴 구성 및 기능을 제공받아야 함
- 3) Component Based UI 활용을 통해 유지보수가 용이하고, 확장성 있는 PJT 환경 구축

### 2. 프로젝트 기술 스택

#### 가. 이슈 관리: Jira

#### 나. 형상 관리: Gitlab

#### 다. 커뮤니케이션: Notion, Mattermost, Slack

#### 라. 개발 환경

- 1) OS: Windows 10
- 2) IDE
  - 가) IntelliJ 2021.3.2
  - 나) Visual Studio Code 1.70.1
  - 다) UI/UX: Figma
- 3) Database:
  - 가) MySQL 8.0.30
  - 나) Redis 7.0.4
- 4) Server: AWS EC2 Ubuntu 20.04 LTS
- 5) Dev-Ops
  - 가) Docker 20.10.17
  - 나) Jenkins 2.60.3

## 마. 상세 사용

### 1) Frontend

- 가) Next.js 12.3.1
- 나) React 18.2.0
- 다) Recoil 0.7.6
- 라) Slick-carousel 1.8.1
- 마) Styled-component 2.8.0
- 바) MUI 5.10.9
- 사) Axios 1.1.3
- 아) React-dropdown 1.11.0
- 자) React-scroll 1.8.8
- 차) React-is-pagination 3.0.3
- 카) React-datepicker 4.8.0
- 타) Prettier 2.7.1
- 파) Moment 2.29.4
- 하) Lodash 4.17.21
- 가) JQuery 3.6.1
- 나) Cors 2.8.5

### 2) Backend

- 가) Spring boot 2.7.2
- 나) Open JDK 8
- 다) Gradle 7.5
- 라) dependencies

```
spring-boot-starter-data-jpa:2.7.5
spring-boot-starter-security:2.7.5
spring-boot-starter-validation:2.7.5
spring-boot-starter-web:2.7.5
mysql-connector-java:8.0.31
Lombok:1.18.24
commons-io:2.6
commons-fileupload:1.4
spring-cloud-starter-aws:2.2.6
```

```
jjwt-api:0.11.2
jjwt-impl:0.11.2
jjwt-jackson:0.11.2
json-simple:1.1.1
spring-boot-starter-mail:1.2.0
spring-boot-starter-data-redis:2.7.5
querydsl-jpa:5.0.0
querydsl-apt:5.0.0
```

### 3. 주요 환경 변수

```
# database
spring.datasource.url=DB 주소
spring.datasource.username=DB 호스트 유저 이름
spring.datasource.password=DB 호스트 유저 비밀번호

# jwt
jwt.secret=jwt 시크릿 키

# s3
cloud.aws.credentials.access-key=발급받은 액세스 키
cloud.aws.credentials.secret-key=발급받은 시크릿 키
cloud.aws.s3.bucket=s3 버킷 이름

# email
spring.mail.username=네이버 아이디
spring.mail.password=네이버 비밀번호

# redis
spring.redis.host=레디스 호스트 주소
spring.redis.port=레디스 포트 번호
spring.redis.password=레디스 비밀번호
```

## 4. 도커 이미지 빌드 및 실행

### 가. Docker 설치

```
sudo apt-get update

sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

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

### 나. Next.js

```
docker build -t frontend .
docker ps -q --filter "name=frontend" | grep -q . && docker stop frontend && docker
rm frontend | true
docker run -d -p 3030:3000 --name frontend frontend
docker ps -q --filter "name=nginx" | grep -q . && docker stop nginx && docker rm
nginx | true
docker run -d -p 80:80 -p 443:443 -v /home/ubuntu/certbot/conf:/etc/letsencrypt/ -v
/home/ubuntu/certbot/www:/var/www/certbot -v
/home/ubuntu/nginx/conf/default.conf:/etc/nginx/conf.d/default.conf --name nginx
nginx:stable-alpine
```

```
docker rmi -f $(docker images -f "dangling=true" -q) || true
```

#### 다. Spring Boot

```
docker build -t backend .  
docker run -p 8081:8080 -d -e TZ=Asia/Seoul --name=backend backend
```

#### 라. Redis

```
docker pull redis  
docker run -d --name redis -p 7000:6379 -v  
/etc/redis/redis.conf:/usr/local/etc/redis/redis.conf redis:latest --requirepass <password>
```

#### 마. Mysql

```
docker pull mysql  
docker run -d --name mysql -e MYSQL_ROOT_PASSWORD=<password> -p 3306:3306  
mysql:latest
```

### 5. 배포 특이사항

#### 가. Spring boot 에 SSL 적용

##### 1) Certbot container 생성 및 인증서 발급

```
sudo mkdir certbot  
cd certbot  
sudo mkdir conf www logs  
  
sudo docker pull certbot/certbot  
sudo docker run -it --rm --name certbot -p 80:80 ₩  
-v "/home/ubuntu/certbot/conf:/etc/letsencrypt" ₩  
-v "/home/ubuntu/certbot/log:/var/log/letsencrypt" ₩  
-v "/home/ubuntu/certbot/www:/var/www/certbot" ₩  
certbot/certbot certonly
```

2) SSL 인증서를 spring boot 에서 필요한 형식(PKCS12)로 변환

```
openssl pkcs12 -export -in fullchain.pem -inkey privkey.pem-out keystore.p12  
-name tomcat -CAfile chain.pem -caname root
```

3) keystore.p12 파일을 /src/main/resources 에 이동

나. nginx 설정

```
upstream front{  
    server {YOUR_HOST};  
}  
  
server {  
    listen 80;  
    server_name k7e201.p.ssafy.io;  
    location / {  
        return 301 https://$host$request_uri;  
    }  
}  
  
server {  
    listen 443 ssl;  
    server_name {YOUR_HOST};  
    access_log /var/log/nginx/access.log;  
    error_log /var/log/nginx/error.log;  
  
    ssl_certificate /etc/letsencrypt/live/{YOUR_HOST}/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/{YOUR_HOST}/privkey.pem;  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 SSLv3;  
    ssl_ciphers ALL;
```

```
location / {  
    proxy_redirect off;  
    charset utf-8;  
  
    proxy_pass http://front;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_set_header X-Forwarded-Proto $scheme;  
    proxy_set_header X-Nginx-Proxy true;  
}  
}
```

## 6.외부 서비스

- 가) [카카오 로그인 기능](#)
- 나) [AWS S3](#)
- 다) [카카오페이 기능](#)