|  |
| --- |
|  |



|  |
| --- |
| **Flowring피뷰티 컨설팅** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |

목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

[4. 외부 서비스 2](#_Toc96072463)

[5. Gitgnore 처리한 핵심 키들 3](#_Toc96072464)

[II. 빌드 3](#_Toc96072465)

[1. 환경변수 형태 3](#_Toc96072466)

[2. 로컬 빌드하기 5](#_Toc96072467)

[3. 배포하기 5](#_Toc96072468)

[4. 서비스 이용 방법 7](#_Toc96072469)

[가) IAMPORT 7](#_Toc96072470)

[나) 소셜로그인 (네이버, 구글, 카카오) 8](#_Toc96072471)

[다) OPEN AI GPT 8](#_Toc96072472)

# 개요

## 프로젝트 개요

연예인 화장법, 유행하는 스타일링. 우리는 언제부터인지 정해진 미의 기준에 자신을 맞춰나가고 있습니다.-

개인마다 주관적인 미를 찾아, 사용자가 자신의 개성을 발견하고 변화하며, 더 나다운 아름다움을 찾을 수 있도록 하는 서비스를 만들고자 하였습니다..

## 프로젝트 사용 도구

이슈 관리 : JIRA, Notion

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 모바비, 프리미어

CI/CD : Jenkins

## 개발환경

VS Code

IntelliJ

JDK : 17 (zulu 17.0.9)

Spring boot : 3.2.0

Node.js : v 20.11.0

SERVER : AWS EC2 Ubuntu 20.04.3 LTS / AWS S3

DB : MariaDB (azure)

## 외부 서비스 및 키 관리

Google OAuth : application-prod.yml에 해당 내용 있음

Naver OAuth : application-prod.yml에 해당 내용 있음

Kakao OAuth : application-prod.yml에 해당 내용 있음

AWS S3 : .env 에 해당 내용 있음  
Redis cloud : application-prod.ym 및 .env에 해당 내용 있음

iamport : application-prod.yml에 해당 내용 있음

Naver Open API : application-prod.yml에 해당 내용 있음

OPEN Ai GPT API : application-prod.yml에 해당 내용 있음(과금 발생 키이기 때문에 취급주의!)

**Key 위치**

React : .env (/front)

Spring : application.yml, application-dev.yml, application-prod.yml

(/back\src\main\resources, 또는 secure-settings에 위치)

# 빌드

## 환경변수 형태

.env

VITE\_APP\_GPT\_API\_KEY=;

VITE\_APP\_API\_KEY= \_I;

VITE\_APP\_AUTH\_DOMAIN=;

VITE\_APP\_PROJECT\_ID=;

VITE\_APP\_STORAGE\_BUCKET=;

VITE\_APP\_MESSAGING\_SENDER\_ID=;

VITE\_APP\_APP\_ID=;

VITE\_APP\_MEASUREMENT\_ID=;

VITE\_APP\_NAVER\_CLIENT\_ID=;

VITE\_APP\_NAVER\_CLIENT\_SECRET=;

VITE\_APP\_BASE\_URL=

VITE\_APP\_ACCESS\_KEY=

VITE\_APP\_SECRET\_KEY=" "

REDIS\_PASSWORD=

Application.yml

spring:  
 profiles:  
 active : dev  
 cache :  
 type : *redis*server:  
 servlet:  
 context-path: /  
 forward-headers-strategy: *framework*

Application-prod.yml

spring:  
 config:  
 activate:  
 on-profile : prod

# face plus plus setting  
 analysis:  
 key: face plus plus key  
 secret: face plus plus secret

# maria DB setting  
 datasource:  
 driver-class-name: org.mariadb.jdbc.Driver  
 url:   
 username:   
 password:   
 hikari:  
 maximum-pool-size: 50  
 minimum-idle: 50  
 connection-timeout: 5000  
 connection-test-query: /\*CONNECTION TEST QUERY\*/SELECT NOW() FROM DUAL  
 idle-timeout: 600000  
 max-lifetime: 1800000  
 auto-commit: true  
 pool-name: DevLogHikariCP

#JPA 설정  
 jpa:  
 hibernate:  
 ddl-auto: none

#redis 설정  
 data:  
 redis:  
 host: redis #Redis Docker 컨테이너 명  
 port: 6379 #Redis 설정 포트번호  
 password : #Redis 비밀번호  
 lettuce:  
 pool:  
 max-active: 8  
 max-idle: 8  
 min-idle: 0  
 max-wait: 2000ms

#mail 설정  
 mail:  
 default-encoding: UTF-8  
 host: smtp.gmail.com  
 port: 587  
 username: # 이메일 서비스에 이용될 이메일 주소  
 password: #설정된 mail 비밀번호 (연결 key)  
 properties:  
 mail:  
 smtp.auth: true  
 smtp.timeout: 50000  
 smtp.starttls.enable: true  
 jwt:  
 access: # JWT ACCESS KEY  
 refresh: # JWT Refresh Key   
 accesstime: # Access key live Time  
 refreshtime: # refresh key live Time  
 security:  
 oauth2:  
 client:  
 registration:  
 kakao:  
 client-id: # Kakao client id  
 client-secret: # Kakao secret key  
 redirect-uri: # redirect url (code 받을 주소)  
 authorization-grant-type: authorization\_code  
 client-authentication-method: client\_secret\_post  
 client-name: Kakao  
 scope: # 카카오 리소스 서버에서 받을 정보들  
 naver:  
 client-id: # Naver client id  
 client-secret: # Naver client secret  
 redirect-uri: # redirect url (code 받을 주소)  
 authorization-grant-type: authorization\_code  
 scope: # 네이버 리소스 서버에서 받을 정보들  
 client-name: Naver  
  
 google:  
 client-id: # Google client id  
 client-secret: # Google secret key  
 redirect-uri: # redirect url (code 받을 주소)  
 scope: profile, email  
# 카카오와 네이버의 설정  
 provider:  
 kakao:  
 authorization-uri: https://kauth.kakao.com/oauth/authorize  
 token-uri: https://kauth.kakao.com/oauth/token  
 user-info-uri: https://kapi.kakao.com/v2/user/me  
 user-name-attribute:   
 naver:  
 authorization-uri: https://nid.naver.com/oauth2.0/authorize  
 token-uri: https://nid.naver.com/oauth2.0/token  
 user-info-uri: https://openapi.naver.com/v1/nid/me  
 user-name-attribute:   
# iamport 설정

iamport:  
 key: #iamport key  
 secret: #iamport secret key

# open ai gpt 설정  
chatgpt :  
 key : # gpt key

## 로컬에서 빌드하기

1) Front

npm install

npm run dev

2) Back-spring

Gradlew build

Java -jar [file-name.jar] --spring.profiles.active=prod

## 배포하기

1. Ec2 인스턴스에 접속
   1. 명령어 : ssh –I mykey.pem (계정)@(ip)
2. Ec2인스턴스에 도커 및 도커 컴포즈 설치
3. Ec2인스턴스에 오픈비두 설치 및 오픈비두 nginx 설정
   1. Docker ps 로 설치가 완료된 도커 컨테이너 확인
   2. Openvidu –proxy 컨테이너에 접속
      1. 명령어 : docker exec –it 컨테이너ID bin/bash

Cd etc/nginx/conf.d

Vi default.conf

https 설정에 service nginx 설정

# Your App

location / {

proxy\_pass [서비스 nginx ip]/;

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\_read\_timeout 600s;

proxy\_connect\_timeout 600s;

}

location /api/ {

proxy\_pass [서비스 nginx ip]/api/;

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-Forwarded-Prefix /api/;

proxy\_read\_timeout 600s;

proxy\_connect\_timeout 600s;

}

nginx –s reload

1. git으로 서비스 소스코드 클론
   1. 소스코드 내 Jenkins docker파일 빌드 후 실행
2. Jenkins 접속 후 빌드 및 배포 자동화 설정 완료

## 서비스 이용 방법

### IAMPORT

1. IAM 포트 공식홈페이지 접속 후 로그인
2. 내 상점 만들기
3. 상점 KEY 발급
4. 사용할 PG사 선택 후 ID 및 KEY 발급

### 소셜로그인(카카오, 네이버, 구글)

1. 카카오, 네이버, 구글의 디벨로퍼 페이지 로그인
2. 어플리케이션 생성
3. 사용할 API 선택
4. Id 와 secret key 발급 -> 도메인 및 redirect url 설정

### OPEN AI GPT

1. gpt 페이지 로그인
2. view api keys 선택
3. 키 발급