|  |
| --- |
|  |

|  |
| --- |
| **Look At me In** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

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

[II. 빌드 3](#_Toc96072465)

[1. 환경변수 형태 3](#_Toc96072466)

[2. 빌드하기 5](#_Toc96072467)

[3. 배포하기 5](#_Toc96072468)

# 개요

## 프로젝트 개요

"LookAtMeIn"은 사용자에게 맞춤형 성형 상담 및 시뮬레이션을 제공하는 인공지능 기반 플랫폼입니다. 이 플랫폼은 개인의 얼굴 사진을 분석하여 성형 전후를 시각화하고, 사용자의 니즈에 맞는 성형 방안을 제안합니다. 현재 성형 상담은 대부분 오프라인에서 이루어지며, 실제 결과를 예측하기 어려운 문제가 있습니다. "LookAtMeIn"은 이러한 문제를 해결하고자, 사용자가 온라인에서 손쉽게 자신에게 맞는 성형 결과를 예측해볼 수 있도록 돕습니다.

#### 타겟 오디언스

* 성형을 고려하고 있는 개인
* 성형 결과에 대한 시뮬레이션을 원하는 이들
* 성형외과 의사 및 클리닉

#### 플랫폼 기능

* 사용자 얼굴 비대칭 분석
* 맞춤형 성형 시뮬레이션
* 성형 화상 상담

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab, Gerrit

커뮤니케이션 : Notion

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins, docker

## 개발환경

Spring: 3.2.1

Java: 17

Gradle: 8.5

JPA: 6.4.2

Pytorch: 1.13.1

Python: 3.10

CUDA: 11.6

FastAPI: 0.103

WebSocket: 6.1.3

Node.js 20.11.0

React 18.0.2

Stomp: 7.0.0

SERVER : AWS EC2 Ubuntu 20.04.3 LTS, OCI Compute Ubuntu 20.04.3 LTS

DB : MySQL

Nginx: 1.25.0

## .gitIgnore 처리한 핵심 키들

React : .env.production(최상단에 위치)

Spring : application.properties

(\src\main\resources 에 위치)

# 빌드

## 환경변수 형태

.env.production

REACT\_APP\_URL // 백엔드 서버로 요청을 보낼 URL

.application.properties

# MySQL DB

spring.datasource.url : DB 주소

spring.datasource.username: 유저이름

spring.datasource.password: 유저 비밀번호

spring.datasource.driver-class-name : com.mysql.jdbc.Driver

# JWT

Jwt.salt : salting을 위한 임의 생성 문자열

Jwt.secret: 시크릿키

## 빌드하기

1) Front

a. npm install --force

b. npm run build

c. dockerfile 작성

FROM nginx

RUN mkdir /app

WORKDIR /app

RUN mkdir ./build

ADD ./build ./build

RUN rm /etc/nginx/conf.d/default.conf

COPY ./nginx.conf /etc/nginx/conf.d

EXPOSE 80

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

d. sh init.sh

docker build --no-cache -t frontend-image .

docker rm frontend-container --force

docker run -p 3000:80 -d --name frontend-container frontend-image

2) Spring

a. sudo chmod +x ./gradlew

b. ./gradlew clean bootJar

c. dockerfile 작성

FROM openjdk:17-ea-slim

ARG JAR\_FILE=./build/libs/lam-0.0.1-SNAPSHOT.jar

COPY ${JAR\_FILE} app.jar

ENTRYPOINT [ "java", "-jar" ,"./app.jar" ]

d. sh init.sh

sudo docker rm spring-container --force

sudo docker build --no-cache -t spring-image .

sudo docker run -p 8080:80 -d --name spring-container spring-image

3) FastAPI

a. [엔비디아그래픽드라이버 565, cuda11.6, cudnn8.9.5 설치](https://gpolya.notion.site/cuda-cudnn-filezila-638d8bd584fc4afbb6c25ca107c07253?pvs=4)

b. 미니콘다 설치해서 conda env create –f environment.yaml –n [env-name]

c. conda activate [env-name]

d. B206-python-repo안에서 python main.py

## 배포하기

### Nginx 설정

#### Ec2에서 lookatmein.conf파일을 작성해서 /etc/nginx/conf.d폴더에 저장

server{

    listen 80;

    server\_name i10b206.p.ssafy.io;

    return 301 https://$host$request\_uri;

}

server {

    listen 443 ssl http2;

    listen [::]:443 ssl http2;

    server\_name i10b206.p.ssafy.io;

    ssl\_certificate /etc/letsencrypt/live/i10b206.p.ssafy.io/fullchain.pem;

    ssl\_certificate\_key /etc/letsencrypt/live/i10b206.p.ssafy.io/privkey.pem;

    location / {

        proxy\_pass http://localhost:3000; # 프론트엔드 서버

        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;

    }

    location /api {

        proxy\_pass http://localhost:8080; # 스프링 서버

        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;

    }

    location /python {

        proxy\_pass https://lookatmein.duckdns.org; # 파이썬 서버

        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;

    }

     location /ws {

        proxy\_pass http://localhost:8080; # WebSocket 서버

        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;

    }

}

#### 이후 sudo systemctl restart nginx