|  |
| --- |
|  |

|  |
| --- |
| GATI |
| 가족 커뮤니티 폐쇄형 SNS |
|  |
|  |
| **포딩 매뉴얼** |
|  |
|  |

목차

[**I. 개요 2**](#_Toc96072459)

[**1. 프로젝트 개요 2**](#_Toc96072460)

[**2. 프로젝트 사용 도구 2**](#_Toc96072461)

[**3. 개발환경 2**](#_Toc96072462)

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

[**II. 빌드 3**](#_Toc96072465)

[**1. 환경변수 형태 3**](#_Toc96072466)

[**2. 빌드하기 5**](#_Toc96072467)

[**3. 배포하기 5**](#_Toc96072468)

[**4. 서비스 이용 방법 7**](#_Toc96072469)

**개요**

1. **프로젝트 개요**

휴대용 모바일 디바이스의 발전에 따른 SNS 이용자 수 증가로 지인들과의 소통이 활발해진 한편 역설적으로 가족간의 소통 부재가 일어나고 있는 현상과 삶의 우선순위로 여타 선진국들이 가족을 중시하는데 반해 한국은 물질적인 풍요를 중시하는 여론 조사 결과를 보며 가족간의 소통 창구가 필요하다 생각했습니다.

가티는 이러한 가족 소통 부재를 해결하기 위한 하나의 창구로써 폐쇄형 SNS입니다. 가티를 통해 소홀했던 가족 관계를 회복하고 물질주의에서 벗어나기를 기대합니다.

1. **프로젝트 사용 도구**

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost, Webex

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins

1. **개발환경**

VS Code

IntelliJ

JVM : 1.8

Gradle : gradle-7.6-bin.zip

Node.js : 16.16.0

Sever : AWS EC2 Ubuntu 22.04.1 LTS

DB : MariaDB

1. **Gitgnore 처리한 핵심 키들**

Vue : .env.loval (최상단 위치)

Spring : application-db.yml

(\src\main\resources, 또는 classPath에 위치)

**빌드**

1. **환경변수 형태**

.env

DB\_USER=DB호스트 유저 정보

DB\_PASS=DB 비밀번호

DB\_URL= DB 주소

.application-db.yml

sever:

port: 포트 번호

spring:

datasource:

username: DB 유저 이름

password: DB 유저 비밀번호

driver-class-name: com.mysql.cj.jdbc.Driver

# 마리아 DB(배포)

datasource:

url: 마리아 DB 주소

driver-class-name: org.mariadb.jdbc.Driver

username: DB 유저 이름

password: DB 유저 비밀번호

# JWT 시크릿 키

auth:

token:

secret-key: 시크릿 키

1. **빌드하기**

1) Front

npm i

npm start

2) Back-spring

Gradle 실행

1. **배포하기**

Nginx 설정

nginx.conf 파일

user www-data;

worker\_processes auto;

pid /run/nginx.pid;

include /etc/nginx/modules-enabled/\*.conf;

events {

worker\_connections 768;

}

http {

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

client\_max\_body\_size 50M;

server\_names\_hash\_bucket\_size 64;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

server {

server\_name ggati.site;

location / {

proxy\_pass http://127.0.0.1:3000;

}

location /api {

proxy\_pass http://127.0.0.1:8080;

}

location /image {

alias /home/ubuntu/image;

}

listen 443 ssl;

}

server {

server\_name i8a805.p.ssafy.io;

location / {

proxy\_pass http://127.0.0.1:3001;

}

location /api {

proxy\_pass http://127.0.0.1:8081;

}

location /image\_dev {

alias /home/ubuntu/image\_dev;

}

listen 443 ssl;

listen [::]:443 ssl;

ssl\_certificate /etc/letsencrypt/live/i8a805.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i8a805.p.ssafy.io/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

}

server {

if ($host = ggati.site) {

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

}

listen 80;

listen [::]:80;

server\_name ggati.site;

location / {

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

}

}

server {

if ($host = i8a805.p.ssafy.io) {

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

}

listen 80;

listen [::]:80;

server\_name i8a805.p.ssafy.io;

location / {

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

}

}

ssl\_prefer\_server\_ciphers on;

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

gzip on;

}