|  |
| --- |
|  |



|  |
| --- |
| **Treeway** |
| **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. GitIgnore 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

현재 창업에 대한 관심이 높아 72.8%의 사람들이 창업을 준비 중이거나

미래에 창업 의향을 가지고 있다.

60대 이상의 기술 기반 창업 기업 수는 해마다 꾸준히 증가하고 있으며,

이는 창업의 높은 인기를 알 수 있다.

그러나 많은 사람들이 무엇을 창업해야 할지, 어떻게 시작해야 할지에 대한 고민이 많으며,

상권 분석을 제대로 하지 않아 사업 부진으로,

창업 후 3년 이내에 폐업을 고민하는 비율이 점차 증가하고 있다.

이러한 문제를 해결하기 위해, 사용자 입력 정보를 바탕으로

빅데이터 기반 상권 분석을 제공하는 프로젝트를 개발하게 되었습니다.

이를 통해 사용자가 원하는 상권 내에서 최적의 매물을 찾을 수 있도록

서비스를 제공하며 창업자에게 도움이 되는 정책과 뉴스,

지역별 트렌드를 파악할 수 있게 도와줍니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins

## 개발환경

VS Code

IntelliJ

Next.js: 14.2.12

Node.js: 20

Java : 17.0.11

Spring Boot: 3.3.3

SERVER : AWS EC2 Ubuntu 20.04.6 LTS

DB : MySQL 8.0.39

## 외부 서비스

KaKaoMap

OpenAPI

Spark

Hadoop

O.Auth2

## Gitignore 처리한 핵심 키들

Spring Boot : application.yml

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

Next.js : .env.production

(root폴더에 위치)

# 빌드

## 환경변수 형태

## Next.js – package.json

{

"name": "treeway",

"version": "0.1.0",

"private": true,

"scripts": {

"dev": "next dev",

"build": "next build",

"start": "next start -p $PORT",

"lint": "next lint"

},

"dependencies": {

"[@ducanh2912/next-pwa](https://npmjs.com/package/@ducanh2912/next-pwa)": "[^10.2.9](https://npmjs.com/package/@ducanh2912/next-pwa)",

"[@motionone/utils](https://npmjs.com/package/@motionone/utils)": "[^10.18.0](https://npmjs.com/package/@motionone/utils)",

"[@reduxjs/toolkit](https://npmjs.com/package/@reduxjs/toolkit)": "[^2.2.7](https://npmjs.com/package/@reduxjs/toolkit)",

"[@tanstack/react-query](https://npmjs.com/package/@tanstack/react-query)": "[^4.36.1](https://npmjs.com/package/@tanstack/react-query)",

"[chart.js](https://npmjs.com/package/chart.js)": "[^4.4.4](https://npmjs.com/package/chart.js)",

"[cookies-next](https://npmjs.com/package/cookies-next)": "[^4.2.1](https://npmjs.com/package/cookies-next)",

"[framer-motion](https://npmjs.com/package/framer-motion)": "[^11.8.0](https://npmjs.com/package/framer-motion)",

"[next](https://npmjs.com/package/next)": "[14.2.12](https://npmjs.com/package/next)",

"[next-pwa](https://npmjs.com/package/next-pwa)": "[^5.6.0](https://npmjs.com/package/next-pwa)",

"[react](https://npmjs.com/package/react)": "[^18](https://npmjs.com/package/react)",

"[react-chartjs-2](https://npmjs.com/package/react-chartjs-2)": "[^5.2.0](https://npmjs.com/package/react-chartjs-2)",

"[react-dom](https://npmjs.com/package/react-dom)": "[^18](https://npmjs.com/package/react-dom)",

"[react-icons](https://npmjs.com/package/react-icons)": "[^5.3.0](https://npmjs.com/package/react-icons)",

"[react-kakao-maps-sdk](https://npmjs.com/package/react-kakao-maps-sdk)": "[^1.1.27](https://npmjs.com/package/react-kakao-maps-sdk)",

"[react-quill](https://npmjs.com/package/react-quill)": "[^2.0.0](https://npmjs.com/package/react-quill)",

"[react-redux](https://npmjs.com/package/react-redux)": "[^9.1.2](https://npmjs.com/package/react-redux)",

"[react-router-dom](https://npmjs.com/package/react-router-dom)": "[^6.26.2](https://npmjs.com/package/react-router-dom)",

"[react-spinners](https://npmjs.com/package/react-spinners)": "[^0.14.1](https://npmjs.com/package/react-spinners)",

"[redux](https://npmjs.com/package/redux" \t "_blank)": "[^5.0.1](https://npmjs.com/package/redux)",

"[redux-persist](https://npmjs.com/package/redux-persist" \t "_blank)": "[^6.0.0](https://npmjs.com/package/redux-persist)",

"[sass](https://npmjs.com/package/sass)": "[^1.79.2](https://npmjs.com/package/sass)",

"[sweetalert2](https://npmjs.com/package/sweetalert2)": "[^11.14.2](https://npmjs.com/package/sweetalert2)"

},

"devDependencies": {

"[@types/node](https://npmjs.com/package/@types/node)": "[^20](https://npmjs.com/package/@types/node)",

"[@types/react](https://npmjs.com/package/@types/react)": "[^18](https://npmjs.com/package/@types/react)",

"[@types/react-dom](https://npmjs.com/package/@types/react-dom)": "[^18](https://npmjs.com/package/@types/react-dom)",

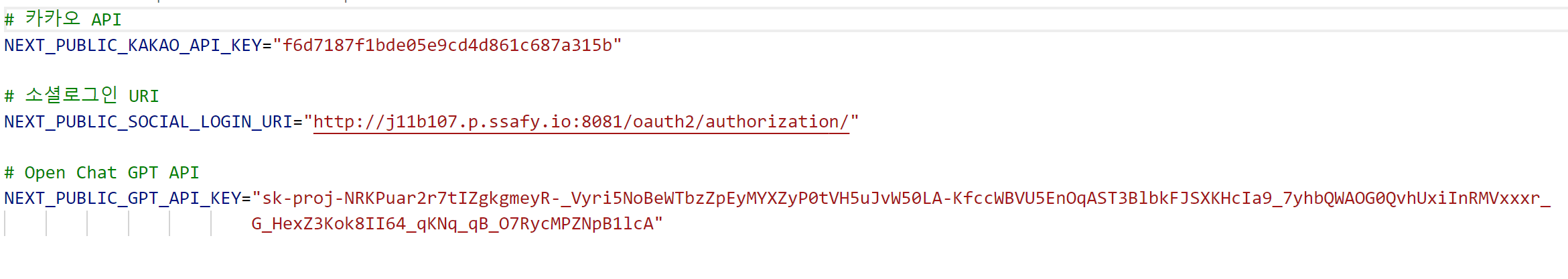
"[quill-image-resize-module-react](https://npmjs.com/package/quill-image-resize-module-react)": "[^3.0.0](https://npmjs.com/package/quill-image-resize-module-react)",

"[typescript](https://npmjs.com/package/typescript)": "[^5](https://npmjs.com/package/typescript)"

}

}

**Next.js -**.**env.production**



**Spring Boot - .application.yml**

spring:  
 servlet:  
 multipart:  
 enabled: true  
 file-size-threshold: 2KB  
 max-file-size: 200MB  
 max-request-size: 215MB  
 security:  
 oauth2:  
 client:  
 registration:  
 google:  
 authorization-grant-type: authorization\_code  
 redirect-uri: https://j11b107.p.ssafy.io/login/oauth2/code/google  
 client-name: google  
 client-id: 778978787532-e6irljoj6d8787dnb2bsos4m1njimaj0.apps.googleusercontent.com  
 client-secret: GOCSPX-yRtQaiQ-19bvOFgwVTNaks-\_ksGf  
 scope: profile,email  
 kakao:  
 client-name: Kakao  
 client-id: 5160c7392d899d15696c86753370747a  
 client-secret: cddyj0xFPx4to987psRfzlAQmMFjVmKH  
 redirect-uri: https://j11b107.p.ssafy.io/login/oauth2/code/kakao  
 authorization-grant-type: authorization\_code  
 client-authentication-method: client\_secret\_post  
 scope:  
 - profile\_nickname  
 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: id  
 jpa:  
 hibernate:  
 ddl-auto: update  
 naming:  
 physical-strategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl  
 properties:  
 hibernate:  
 format\_sql: 'true'  
 show-sql: 'true'  
 datasource:  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 password: 'b107'  
 username: ssafy  
 url: jdbc:mysql://j11b107.p.ssafy.io:3306/tree\_way  
 application:  
 name: treeway  
 jwt:  
 secret: vmfhaltmskdlstkfkdgodyroqkfwkdbalroqkfwkdbalaaaaaaaaaaaaaaaabbbbb  
server:  
 *# address: localhost* port: '8080'  
custom:  
 redirect-url: http://j11b107.p.ssafy.io

**docker-compose.yml**

version: '3.8'

services:

backend:

image: treeway-dev-backend:latest

container\_name: treeway-dev-backend

ports:

- 8081:8080

environment:

- TZ=Asia/Seoul

volumes:

- /etc/localtime:/etc/localtime:ro

- /etc/timezone:/etc/timezone:ro

networks:

- treeway-network

frontend:

image: treeway-dev-frontend:latest

container\_name: treeway-dev-frontend

ports:

- 3001:80

environment:

- TZ=Asia/Seoul

volumes:

- /etc/localtime:/etc/localtime:ro

- /etc/timezone:/etc/timezone:ro

networks:

- treeway-network

networks:

treeway-network:

driver: bridge

**dockerFile – frontend**

# 1단계 애플리케이션 빌드

# Node.js 베이스 이미지를 사용하여 애플리케이션을 빌드

FROM node:20 AS build

# 작업 디렉토리를 /app으로 설정

WORKDIR /app

# package.json과 package-lock.json을 복사하여 의존성을 설치

COPY package\*.json ./

# 필요한 모든 의존성을 설치

RUN npm install

# 프론트엔드 밑에 모든 파일을 복사

COPY . .

# Next.js 애플리케이션을 빌드

RUN npm run build

# 2단계 실행환경 설정

FROM node:20-alpine

# 작업 디렉토리를 /app으로 설정

WORKDIR /app

# build 스테이지에서 빌드된 파일을 복사

# COPY --from=build /app/.next ./.next

# COPY --from=build /app/node\_modules ./node\_modules

# COPY --from=build /app/package\*.json ./

# COPY --from=build /app/public ./public

# COPY --from=build /app/next.config.mjs ./next.config.mjs

COPY --from=build /app /app

# 환경 변수 설정

ENV PORT=3001

# 포트 설정 (Next.js는 기본적으로 3000번 포트 사용)

EXPOSE 3001

# Next.js 애플리케이션을 시작하는 명령어

CMD ["npm", "run", "start"]

**Jenkinsfile – frontend**

pipeline {

agent any

environment {

TARGET\_BRANCH='develop/fe'

}

stages {

stage('Checkout') {

steps {

script {

// def branch = env.gitlabSourceBranch ?: 'develop/fe'

// if(branch != env.TARGET\_BRANCH) {

// error "This build is only for the '${env.TARGET\_BRANCH}' branch, but was run on '${branch}'"

// }

checkout scm

}

}

}

stage('Set .ENV File') {

steps {

dir('frontend/treeway') {

sh 'rm -f .env.production'

sh 'rm -f .env.production.swp'

withCredentials([file(credentialsId: '.ENV.PRODUCTION', variable: 'ENV\_PRODUCTION')]) {

sh 'cat $ENV\_PRODUCTION > .env.production'

}

}

}

}

stage('Build Docker Image') {

steps {

script {

sh 'docker build --no-cache -t treeway-dev-frontend:latest ./frontend/treeway'

// sh 'docker build -t treeway-dev-frontend:latest ./frontend'

}

}

}

// stage('Stop old Container') {

// steps {

// script {

// sh 'docker stop treeway-dev-frontend'

// sh 'docker rm treeway-dev-frontend'

// }

// }

// }

stage('Depoly') {

steps {

dir('/var/jenkins\_home/workspace/test2') {

sh 'docker-compose stop frontend || true'

sh 'docker-compose rm -f frontend || true'

sh 'docker-compose up frontend -d'

}

}

}

stage('Remove old image') {

steps {

script {

sh 'docker image prune -f'

}

}

}

}

}  
  
**dockerfile – backend**

FROM amazoncorretto:17-alpine AS build

WORKDIR /app

# 의존성 설치 및 빌드

COPY build.gradle settings.gradle gradlew ./

COPY gradle ./gradle

COPY src ./src

RUN ./gradlew build --no-daemon

# 빌드 결과물을 실행 이미지로 복사

FROM amazoncorretto:17-alpine

WORKDIR /app

COPY --from=build /app/build/libs/treeway-0.0.1-SNAPSHOT.jar /app/treeway-0.0.1-SNAPSHOT.jar

#EXPOSE 8081

CMD ["java", "-jar", "/app/treeway-0.0.1-SNAPSHOT.jar"]

**Jenkinsfile – backend**

pipeline {

agent any

// environment {

// TARGET\_BRANCH='develop/be'

// }

stages {

stage('Checkout') {

steps {

script {

// def branch = env.gitlabSourceBranch ?: 'develop/be'

// if(branch != env.TARGET\_BRANCH) {

// error "This build is only for the '${env.TARGET\_BRANCH}' branch, but was run on '${branch}'"

// }

checkout scm

}

}

}

stage('Build') {

steps {

dir('backend/treeway') {

withCredentials([file(credentialsId: 'APPLICATION\_YML', variable: 'application\_yml')]) {

sh 'cp $application\_yml ./src/main/resources/application.yml'

}

sh 'chmod +x ./gradlew'

sh './gradlew clean build --no-daemon'

// sh './gradlew test -i'

}

}

}

stage('Build Docker Image') {

steps {

script {

// def image = docker.build("treeway-dev-backend:latest", 'backend/.')

// image.push()

sh 'docker build -t treeway-dev-backend:latest ./backend/treeway'

}

}

}

// stage('Stop old Container') {

// steps {

// script {

// sh 'docker stop treeway-dev-backend'

// sh 'docker rm treeway-dev-backend'

// }

// }

// }

stage('Depoly') {

steps {

dir('/var/jenkins\_home/workspace/test') {

sh 'docker-compose stop backend || true'

sh 'docker-compose rm -f backend || true'

sh 'docker-compose up backend -d'

// sh 'docker-compose up -d'

}

}

}

stage('Remove old Image') {

steps {

script {

sh 'docker image prune -f'

}

}

}

}

}

## 빌드하기

1) Next.js

npm i

npm run dev

2) Spring Boot

Gradle 실행

Bootjar 실행

## 배포하기

Nginx 설정

Default.conf

upstream jenkins {

# server compose-jenkins:9090;

server localhost:9090;

}

upstream frontend {

server 172.22.0.2:3001;

}

upstream backend {

server localhost:8081;

}

server {

listen 80;

listen [::]:80;

server\_name j11b107.p.ssafy.io;

#access\_log /var/log/nginx/host.access.log main;

location / {

#root /usr/share/nginx/html;

#index index.html index.htm;

rewrite ^(.\*) https://j11b107.p.ssafy.io:443$1 permanent; #https redirect

}

############

#SSL #

############

location /.well-known/acme-challenge {

default\_type "text/plain";

root /var/validation;

try\_files $uri $uri/ =404;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root /usr/share/nginx/html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

server {

listen 443 ssl; #ip v4

listen [::]:443 ssl; #ip v6

server\_name j11b107.p.ssafy.io;

#ssl on;

#ssl\_certificate /etc/ssl/certificate.crt;

#ssl\_certificate\_key /etc/ssl/private.key;

ssl\_certificate /etc/letsencrypt/live/j11b107.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/j11b107.p.ssafy.io/privkey.pem;

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

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

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

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

location / {

proxy\_pass http://frontend;

}

###########

# Jenkins #

###########

location /jenkins {

proxy\_pass http://jenkins;

}

###########

# Backend #

###########

location /api {

proxy\_pass http://backend;

}

# Proxy

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-Proto https;

proxy\_headers\_hash\_bucket\_size 512;

proxy\_redirect off;

# Websockets

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

}