|  |
| --- |
|  |





|  |
| --- |
| 하우스코어 |
| :내 집 추천 서비스 |
| 포팅 매뉴얼 |
|  |
|  |
|  |
|  |

목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 기술 스택 2](#_Toc96072461)

[3. 외부 서비스 3](#_Toc96072463)

[4. Gitignore 3](#_Toc96072464)

[II. 빌드 4](#_Toc96072465)

[1. Gitlab에서 프로젝트 받아오기 4](#_Toc96072466)

[2. 환경변수 형태 4](#_Toc96072467)

[3. 프로젝트 빌드하기 4](#_Toc96072467)

[4. 배포하기 5](#_Toc96072468)

[5. 서비스 이용 방법 6](#_Toc96072469)

[가) 백엔드 프로젝트 최상단Docker Compose.yaml 실행을 통해 컨테이너 빌드 6](#_Toc96072470)

[나) (선택) 자동화 파이프라인 구축 시 CI/CD 툴 추가 7](#_Toc96072471)

[다) Mobile 폴더 내 프로젝트 최상단에 명령어 입력으로 mobile apk 빌드 후 실행](#_Toc96072472)

[7](#_Toc96072472)

# 개요

## 프로젝트 개요

**“AI 거주지 평가 시스템”**

‘하우스코어(HOUSCORE)’는 공공 데이터를 활용한 AI 거주지 (건물) 평가 시스템으로 사용자에게 거주지에 대한 평가를 제공합니다.

공인중개사들의 설문을 바탕으로 건물 정보, 인프라 시설, 교통, 치안, 실거래가 공공 데이터를 수집하여 사용자에게 해당 거주지의 AI 점수를 제공, 선택에 도움이 될 수 있도록 합니다.

원하는 거주지의 주소를 입력 하고 해당 거주지의 상세 정보와 지표 정보를 검색해보세요. AI 점수를 통해 해당 거주지의 객관적인 지표를 확인하고 실거주자의 평가 점수와 리뷰를 비교해 거주지를 선택할 수 있습니다.

## 프로젝트 기술 스택

## 1) 모바일

## 언어: Dart 3.3.3

## 프레임워크: Flutter 3.19.5

## 라이브러리: dio ^5.0.3, retrofit ^4.0.1, flutter\_riverpod ^1.0.4, go\_router ^7.0.1, image\_picker ^1.1.0

## 개발환경: Android Studio Iguana | 2023.2.1 Patch 1

## 그 외: Figma

## 2) 백엔드

## 언어: Java 17

## 프레임워크: Springboot 3.2.4, SpringSecurity 6.2.3, JPA 3.2.4, Batch 5.1.1

## 라이브러리: Lombok 1.18.30, Swagger 4.15.5

## 개발환경: IntelliJ IDEA 2024.1

## 그 외: S3

## 3) AI

## 언어: Python 3.11

## 라이브러리: FastAPI 0.111.0, pandas 2.1.4, pycaret 3.3.1, scikit-learn 1.4.2, xgboost 2.0.3, lightgbm 4.3.0

## 개발환경: PyCharm Community Edition 2023.3.3, anaconda 23.7.4

**4) 협업 툴**

이슈 관리 : JIRA

형상 관리 : Gitlab, Docker 25.0.4

커뮤니케이션 : Notion, Mattermost, Gerrit

**5) 인프라**

서버 환경: Amazon EC2 Ubuntu 20.04.6 LTS

웹 서버: Nginx 1.18.0

**6) CI/CD**

배포 자동화 파이프라인: Jenkins jdk17

아티팩트 저장소: Sonatype Nexus 3.66.0

정적 분석: SonarQube 4.4.1

**7) Database**

PostgreSQL 16.2, MongoDB 7.0.8

**3. 외부 서비스**

주소검색&소셜로그인: KAKAO API

## 4. Gitignore

Spring : application.properties

(Backend/SpringbootApp/houscore/src/main/resources 경로에 위치)

Python : .env

(ai 경로에 위치)

# 빌드

## 1. Gitlab에서 프로젝트 받아오기 (프로젝트 접근 토큰 필요)

git clone https://lab.ssafy.com/s10-final/S10P31S206.git

git checkout master

## 2. 환경변수 형태

**.env**

//Database 접근 포트 및 키

REDIS\_HOST=  
REDIS\_PORT=  
REDIS\_PASSWORD=   
  
POSTGRES\_DB=   
POSTGRES\_PASSWORD=

POSTGRES\_USER=   
POSTGRES\_PORT=

//서버 포트  
SERVER\_PORT=

**application.properties**

*DB\_URL* =   
*DB\_USERNAME* =   
*DB\_PASSWORD* =   
  
*MONGO\_URL* =   
  
*KAKAO\_ID* =   
*KAKAO\_SECRET* =   
  
*S3\_ACCESS\_KEY* =   
*S3\_SECRET\_KEY* =   
*SERVER\_PORT* =   
  
*FASTAPI\_BASE\_URL* =

**.env (파이썬)**

*POSTGRES\_HOST=*

*POSTGRES\_DB=*

*POSTGRES\_PASSWORD=*

*POSTGRES\_USER=*

*POSTGRES\_PORT=*

*MATTERMOST\_WEB\_HOOK\_URL=*

## 프로젝트 빌드하기

1) Mobile

IOS App store에 어플리케이션 출시

2) Backend-FastAPI

- Dockerfile 위치 : ai 경로에 위치

Docker build –t houscore-ai .

.

3) Backend-Spring

- Gradle build실행

./gradlew build

- 빌드된 프로젝트를 Dockerfile화

docker build -t houscore-springboot-app .

## 배포하기

**Nginx 설정(certbot 사용)**

# Default server configuration

server {

listen 80;

listen [::]:80;

server\_name k10s206.p.ssafy.io;

return 301 https://$host;

}

server {

listen [::]:443 ssl ipv6only=on; # managed by Certbot

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/k10s206.p.ssafy.io/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/k10s206.p.ssafy.io/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

server\_name k10s206.p.ssafy.io; # managed by Certbot

location / {

proxy\_pass http://localhost:8080/;

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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /oauth2/ {

proxy\_pass http://localhost:8084/oauth2/;

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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

}

location /login/ {

proxy\_pass http://localhost:8084/login/;

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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

}

location /api/ {

proxy\_pass http://localhost:8084/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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

}

location /portainer/ {

proxy\_pass http://localhost:9000/;

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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

}

location /sonarqube/ {

proxy\_pass http://localhost:9001/;

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\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

}

}

## 서비스 이용 방법

### 백엔드 프로젝트 최상단Docker Compose.yaml 실행을 통해 컨테이너 빌드

version: '3.8'  
services:  
 app:  
 image: houscore-springboot-app  
 ports:  
 - "8084:8084"  
 environment:  
 - SPRING\_PROFILES\_ACTIVE=prod  
 ai\_service:  
 image: houscore-fastapi-app  
 ports:  
 - "8000:8000"

### (선택) 자동화 파이프라인 구축 시 CI/CD 툴 추가

Jenkins: 자동화 파이프라인 구축 및 깃랩 연동

매뉴얼: https://velog.io/@bzeromo/%EC%8E%84%ED%8A%B8%EB%A0%89%EC%95%84%EC%9D%B4-Jenkins

Sonatype nexus Repository: .env 파일 등 보안이 필요한 파일 및 빌드 파일 보관

매뉴얼: <https://velog.io/@bzeromo/DevOps-Artifact-repository-Sonatype-Nexus-Repository>

SonarQube: 코드 품질에 대한 정적 분석

매뉴얼: <https://velog.io/@bzeromo/DevOps-SonarQube>

### 다) Mobile 폴더 내 프로젝트 최상단에 명령어 입력으로 mobile apk 빌드 후 실행

Flutter build apk --release

.