우리가족 거우리

포팅 매뉴얼

**소속**\_서울 2반 1팀

**지도 교수**\_유상진 컨설턴트님

**프로젝트명**\_우리가족 거우리

**제작**\_서보민/노현정/이동민/원영현/심정윤/유소영

목차

1. 프로젝트 기술 스택
2. Embedded
3. Frontend
4. Backend
5. 빌드 및 배포
6. 빌드 설정 파일
7. 외부 서비스
8. Firebase Storage
9. Teachable Machine

# 프로젝트 기술 스택

## Embedded

* + 1. 개발환경
       1. Python : 3.8.10
       2. Ubuntu : 20.04
       3. node : 16.19.0
       4. npm : 8.19.3
    2. 사용 기술
       1. 초음파센서 - 아두이노
       2. 카메라
          1. 얼굴인식 - OpenCV
          2. 동작인식 - Teachable Machine
       3. 마이크
          1. STT - gSTT
          2. 감정 분석 - gNLP
       4. 스피커
          1. TTS - gTTS
       5. Web socket
    3. 스마트미러장
       1. 라떼판다
       2. 슬라이드거울장
       3. 아크릴판
       4. 하프미러필름
       5. 24인치 모니터
       6. 초음파센서
       7. 카메라
       8. 마이크
       9. 스피커wd

## Frontend

* + 1. 개발환경
       1. node : 16.19.0
       2. npm : 8.19.3
       3. React.js
       4. yarn : 1.22.19
    2. 사용 기술
       1. web socket
       2. Redux-toolkit
       3. React-router
       4. mui
       5. axios
       6. react bootstrap

## Backend

* + 1. 개발환경
       1. Java8 zulu 8.66.0.15
       2. IntelliJ 2022.3
       3. Spring Boot 2.7.7
       4. Gradle-Kotlin
    2. DB
       1. mariaDB
       2. firebase Storage
    3. 사용 기술, 라이브 러리
       1. Spring JPA
       2. Lombok
       3. CoolSms API
       4. ical4j : 4.0.0-beta2
       5. jsoup : 1.15.3
       6. Docker : 20.10.23
       7. Nginx

# 빌드 및 배포

## 빌드 과정

**Spring Boot project**

1. **프로젝트 루트 경로 진입**
2. **./gradlew build**
3. **루트 경로에서 → /build/libs 경로 진입**
4. **java -jar SmartMirror-0.0.1-SNAPSHOT.jar 로 jar 파일을 실행합니다.**

**React project**

1. **프로젝트 루트 경로 진입**
2. **npm install**
3. **install 완료**
4. **npm run build**
5. **npm run build 완료 후 docker image 파일을 만들어 container화 시켜 사용하였습니다.**

## 빌드 및 배포 파일 내용

**Dockerfile**

* **Spring Boot\_Dockerfile**
* **FROM** openjdk:8-jre
* **COPY** build/libs/SmartMirror-0.0.1-SNAPSHOT.jar app.jar
* **ENTRYPOINT** ["java", "-jar", "app.jar"]
* **React\_Dockerfile**
* **FROM** node:alpine
* **COPY** package.json ./
* **RUN** npm install
* **COPY** ./ ./
* **CMD** ["npm","run","start"]

**application.properties**

*# mariaDB*

spring.datasource.driverClassName=org.mariadb.jdbc.Driver

spring.datasource.url=jdbc:mariadb://stg-yswa-kr-practice-db-master.mariadb.database.azure.com:3306/S08P12A201?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8

spring.datasource.username=S08P12A201

spring.datasource.password=\*\*\*\*

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.show-sql=true

*# firebase*

app.firebase-configuration-file=./serviceAccountKey.json

app.firebase-bucket=ddok-mirror.appspot.com

*# file upload max size*

spring.servlet.multipart.maxFileSize=10MB

spring.servlet.multipart.maxRequestSize=10MB

**NginX 설정**

**default.conf**

upstream backend {

server i8a201.p.io:8081;

}

upstream frontend {

server i8a201.p.io:3000;

}

server {

listen 80;

listen [::]:80;

server\_name i8a201.p.io;

location / {

proxy\_pass http://frontend;

proxy\_set\_header Connection "";

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /web {

proxy\_pass http://backend;

proxy\_set\_header Connection "";

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /mirror {

proxy\_pass http://backend;

proxy\_set\_header Connection "";

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

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

location = /50x.html {

root /usr/share/nginx/html;

}

}

# 외부 서비스

## Firebase Storage

거울에 부착된 카메라를 이용해 촬영한 사진을 업로드 하기 위한 저장소로 Firebase Storage를 사용

1. Firebase 회원가입
2. Firebase project 생성
3. Firebase Database 생성
4. Firebase Cloud Storage 생성 및 버킷 위치 설정
5. Spring Boot 프로젝트의 gradle dependency 추가
   1. implementation("com.google.firebase:firebase-admin:9.1.1")
6. 아래 코드를 이용해 uploadFiles 사용

@Service

**public** **class** **FireBaseInitializer** {

@Value("${app.firebase-configuration-file}")

**private** String firebaseConfigPath;

@PostConstruct

**public** void initialize(){

**try**{

FirebaseOptions options = **new** FirebaseOptions.Builder().setCredentials(

GoogleCredentials.fromStream(

**new** ClassPathResource(firebaseConfigPath).getInputStream())).build();

**if**(FirebaseApp.getApps().isEmpty()){

FirebaseApp.initializeApp(options);

}

} **catch** (IOException e) {

}

}

}

@Service

**public** **class** **FireBaseService** {

@Value("${app.firebase-bucket}")

**private** String firebaseBucket;

**public** String uploadFiles(MultipartFile file, String nameFile) **throws** IOException {

Bucket bucket = StorageClient.getInstance().bucket(firebaseBucket);

InputStream content = **new** ByteArrayInputStream(file.getBytes());

Blob blob = bucket.create(nameFile.toString(), content, file.getContentType());

String fileUrl = "https://firebasestorage.googleapis.com/v0/b/" + firebaseBucket +"/o/" + nameFile + "?alt=media";

**return** fileUrl;

}

}

## Teachable Machine

아이가 양치를 하는 과정에서 잘 하고 있는지, 자리를 비웠는지 여부를 파악하기 위해 Teachable Machine을 사용하였습니다.