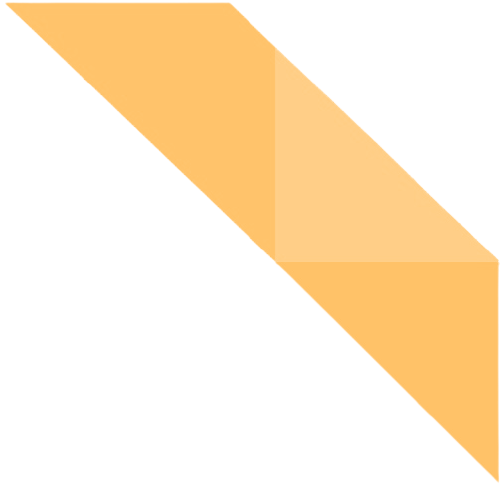
|  |
| --- |



| **아이링크** |
| --- |
| **포팅 매뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[**I. 개요 2**](#_1fob9te)

[**1. 프로젝트 개요 2**](#_3znysh7)

[**2. 프로젝트 사용 도구 2**](#_2et92p0)

[**3. 개발환경 2**](#_tyjcwt)

[**4. 외부 서비스 2**](#_3dy6vkm)

[**5. Gitgnore 처리한 핵심 키들 3**](#_1t3h5sf)

[**II. 빌드 3**](#_4d34og8)

[**1. 환경변수 형태 3**](#_2s8eyo1)

[**2. 빌드하기**](#_17dp8vu) 3

[**3. 배포하기**](#_3rdcrjn)5

# 개요

## 프로젝트 개요

부모님들이 아이들을 보육시설에 보낼 시기가 되면 가장 먼저 고려하는 부분은 원과의 소통입니다.

아이링크는 그러한 니즈를 만족하기 위해 편리하게 정보 공유를 할 수 있는 환경을 제공하고 키오스크를 활용한 아이의 참여를 통해 원과 가정, 아이를 이어주는 서비스입니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

Open API 관리 : Swagger

커뮤니케이션 : Notion, Miro Board, Discord

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins

## 개발환경

Text Editor: Visual Studio Code IDE

Backend: Express

Frontend: React

SERVER : AWS EC2 Ubuntu 20.04 LTS, Jenkins, NGINX, PM2

DB : MySQL, Redis

## 외부 서비스

다음 주소 api : src/components/Center/searchform.jsx (유치원 주소등록시 사용)

날씨 API : /front/i\_link/src/components/Weather/index.jsx (날씨 데이터, 아이콘 표시)

크롬 TTS : front/i\_link/src/components/TTSButton/index.jsx (브라우저 TTS)

## Gitgnore 처리한 핵심 키들

Backend(Node.js) : .env (back/app/.env)

# 빌드

## 환경변수 형태

.env

#MySQL

DB\_NAME = “i\_link”

DB\_USER = “MySQL 호스트 유저 정보”

DB\_PASS = “MySQL 호스트 유저 비밀번호”

DB\_HOST = “MySQL 주소”

#JWT

JWT\_ACCESS\_SECRET = “JWT ACCESS 시크릿키”

JWT\_REFRESH\_SECRET = “JWT REFRESH 시크릿키”

JWT\_ACCESS\_TIME = “JWT ACCESS Key 만료시간”

JWT\_REFRESH\_TIME = “JWT REFRESH Key 만료시간”

#REDIS

REDIS\_HOST = “REDIS 주소”

REDIS\_PORT = “REDIS Port”

REDIS\_PASS = “REDIS Password”

## 빌드하기

0) 공통

1. AWS Ubuntu 환경 구성
2. 도커 설치
3. letsencrypt로 도메인 SSL 인증
4. Ubuntu홈디렉토리/env 위치에 .env 파일 작성 ( 환경변수 형태 참고 )
5. git clone
6. Project Root 디렉토리로 이동

1) Front

./front/i\_link/config/default.conf 서버도메인 -> 본인 도메인으로 변경

server {

listen 80;

server\_name 서버도메인;

location / {

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

}

}

server {

listen 443 ssl;

server\_name 서버 도메인;

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

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

ssl\_certificate /etc/letsencrypt/live/서버도메인/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/서버도메인/privkey.pem;

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2 SSLv3;

ssl\_ciphers ALL;

location / {

root /usr/share/nginx/html;

index index.html index.htm

proxy\_redirect off;

charset utf-8;

try\_files $uri $uri/ /index.html;

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;

proxy\_set\_header X-Nginx-Proxy true;

}

}

default.conf 파일 수정 완료 후

docker build -t i\_link ./front/i\_link/

2) Back

docker build -t i\_link\_back ./back/app/

## 배포하기

0) 공통

Project Root 디렉토리로 이동

1) Front

docker run --name i\_link -d -p 80:80 -p 443:443 -v {letsencrypt SSL 인증 키 디렉토리}:/etc/letsencrypt i\_link

2) Back

docker run --name i\_link\_back -v {Ubuntu 홈 디렉토리}/env/.env:/usr/src/app/.env -v {letsencrypt SSL 인증 키 디렉토리}:/etc/letsencrypt -v {Ubuntu 홈 디렉토리}/uploads:/usr/src/app/uploads -d -p 8000:8000 i\_link\_back