



맘대로: 온라인 심리 상담 서비스

포팅 메뉴얼

목차

- I. 개요 2
 - 1. 프로젝트 개요 2
 - 2. 프로젝트 사용 도구 2
 - 3. 개발 도구 2
 - 4. 개발 환경 2
 - 5. 외부 서비스 3
- II. 빌드 4
 - 1. 환경변수 4
 - 2. CI/CD 구축 7
 - 3. 배포 12
- III. 시연 시나리오 13

I. 개요

1. 프로젝트 개요

“맘대로”는 개인의 정신 건강을 지원하기 위한 온라인 심리 상담 서비스입니다. 이 서비스는 사용자가 쉽게 접근할 수 있는 심리 상담 및 지원을 제공하며, 전문 심리 상담사와의 1:1 상담, 자가 심리 검진, 감정 일기를 통해 정신적 웰빙을 증진하고 스트레스를 관리하는 데 도움을 주는 것을 목표로 합니다.

2. 프로젝트 사용 도구

이슈 관리 : JIRA, Gitlab

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

ERD: ERD Cloud

디자인 : Figma

CI/CD : Jenkins

UCC :

3. 개발 도구

VS Code : 1.90.2

IntelliJ : 2024.1.4

HeidiSQL : 12.8.0.6908

4. 개발 환경

Frontend

Node.js 20.15.0

React 18.3.1

vite 5.3.4

react-quey 5.51.11

axios 1.7.3

zustand 4.5.4

openai 4.55.1

Backend

Java 17

Spring Boot 3.3.1

Gradle 8.8

Server

AWS EC2 Ubuntu 20.04.6 LTS

Service

Mysql 8.0.38

Nginx 1.18.0

Docker 27.1.1

Jenkins 2.469

5. 외부 서비스

채팅 OpenAI API Key : /properties/.env 에 해당 내용 있음 -> front/ 폴더에 넣어야함

STT 요약 OpenAi API Key : /properties/application-secret.yml 에 해당 내용 있음

STT CLOVA API Key : /properties/application-secret.yml 에 해당 내용 있음

이미지 저장 Aws S3 Key : /properties/application-secret.yml 에 해당 내용 있음

메일 발송 Mail : /properties/application-secret.yml 에 해당 내용 있음

Livekit API Key : /properties/application.yml 에 해당 내용 있음

II. 빌드

1. 환경변수

1) .env

```
VITE_OPENAI_API_KEY =  
VITE_PORT = 5173
```

2) application.yml

```
spring:  
  application:  
    name: mamdaero  
  profiles:  
    include: secret  
  datasource:  
    driver-class-name: com.mysql.cj.jdbc.Driver  
    hikari:  
      maximum-pool-size: 100  
      idle-timeout: 600000  
      max-lifetime: 1800000  
      connection-timeout: 30000  
      leak-detection-threshold: 3000  
  jpa:  
    open-in-view: false  
    properties:  
      hibernate:  
        show_sql: true  
        format_sql: true  
        ddl-auto: create  
    database-platform: org.hibernate.dialect.MySQLDialect  
  file:  
    upload-dir: /home/ubuntu/audio-save
```

3) application-secret.yml

```
spring:
  datasource:
    username: root
    password:
  servlet:
    multipart:
      max-file-size: 100MB
      max-request-size: 100MB
  mail:
    username:
    password:

  jwt:
    secret:

  access:
    expiration: 36000
    header: Authorization

  refresh:
    expiration: 72000
    header: Authorization-refresh

# S3
cloud:
  aws:
    s3:
      bucketName: mamdaero
      path:
        profile: profile/
        board: board/
  credentials:
    accessKey:
```

```
    secretKey:
    region.static:
    stack.auto:

# STT
CLOBA:
    SECRET:
    INVOKE_URL:

# ChatGPT
OPENAI:
    SECRET:

thymeleaf:
    prefix: classpath:/mail-templates/
    suffix: .html
    mode: HTML
    encoding: UTF-8
    check-template-location: true
    cache: false
```

2. CI/CD 구축

- 1) EC2 에 Docker 설치하기
- 2) Docker 이미지, 컨테이너 생성 – Jenkins, Mysql
 - A. Jenkins : `docker run -d -p 9090:9090 -v /home/ubuntu/jenkins-data:/var/jenkins_home --name jenkins jenkins/jenkins:jdk17`
 - B. Mysql : `docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=비밀번호 -v /home/ubuntu/mysql:/var/lib/mysql --name mysql2 mysql:8.0.38`

- 3) Backend DockerFile 작성

```
FROM openjdk:17-jdk-alpine
```

```
ARG JAR_FILE=build/libs/*.jar
```

```
COPY ${JAR_FILE} app.jar
```

```
EXPOSE 8080
```

```
ENTRYPOINT ["java","-jar","/app.jar"]
```

- 4) Docker 이미지, 컨테이너 생성 – Nginx
- 5) SSL 인증서 발급
- 6) Nginx – nginx.conf 작성

```
server {  
    listen 80;  
    server_name mamdaero;  
  
    location /jenkins/ {  
        proxy_pass http://localhost:8081/  
        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;  
    }  
  
    # Redirect HTTP to HTTPS  
    return 301 https://$host$request_uri;  
}  
  
server {  
    listen 443 ssl;
```



```
server_name mamdaero;

charset utf-8;

ssl_certificate /etc/letsencrypt/live/mamdaero.o-r.kr/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/mamdaero.o-r.kr/privkey.pem;

# SSL settings
ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_ciphers HIGH:!aNULL:!MD5;

location = /api/ {
    root /usr/share/nginx/html;
    index index.html;
    try_files $uri /index.html;
}

# Reverse proxy for /api to the Spring Boot application
location /api/ {

    proxy_pass http://spring: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_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_buffering off;
    proxy_cache off;
    chunked_transfer_encoding off;
    break;
}
```

```
# Serve React application for other requests
location / {
    root /usr/share/nginx/html;
    index index.html;
    try_files $uri $uri/ /index.html;
}
}
```

7) DockerFile 작성

```
FROM nginx:latest

USER root

COPY nginx.conf /etc/nginx/conf/nginx.conf
COPY /etc/letsencrypt/live/mamdaero.o-r.kr/fullchain.pem
/etc/letsencrypt/live/mamdaero.o-r.kr/fullchain.pem
COPY /etc/letsencrypt/live/mamdaero.o-r.kr/privkey.pem
/etc/letsencrypt/live/mamdaero.o-r.kr/privkey.pem

EXPOSE 80 443
```

8) docker-compose.yml 작성

```
services:
  nginx:
    image: nginx:latest
    container_name: nginx
    volumes:
      - /home/ubuntu/nginx-docker/nginx.conf:/etc/nginx/conf.d/default.conf
      - /etc/letsencrypt/live/mamdaero.o-
r.kr/fullchain.pem:/etc/letsencrypt/live/mamdaero.o-r.kr/fullchain.pem
```

```
- /etc/letsencrypt/live/mamdaero.o-  
r.kr/privkey.pem:/etc/letsencrypt/live/mamdaero.o-r.kr/privkey.pem  
- /home/ubuntu/mamdaero2/frontend/dist:/usr/share/nginx/html  
ports:  
- "80:80"  
- "443:443"  
depends_on:  
- spring  
networks:  
- internal_network
```

mysql:

image: mysql:8.0.38

container_name: mysql2

environment:

- MYSQL_DATABASE=\${MYSQL_DATABASE}
- MYSQL_ROOT_HOST=\${MYSQL_ROOT_HOST}
- MYSQL_ROOT_PASSWORD=\${MYSQL_ROOT_PASSWORD}
- TZ=Asia/Seoul

command:

```
[  
  "--character-set-server=utf8mb4",  
  "--collation-server=utf8mb4_unicode_ci",  
]
```

ports:

- "3306:3306"

volumes:

- /home/ubuntu/mysql:/var/lib/mysql

networks:

- internal_network

spring:

build: /home/ubuntu/mamdaero2/backend

restart: always

ports:

```
- "8080:8080"
depends_on:
  - mysql
container_name: spring
environment:
  - SPRING_DATASOURCE_URL=${SPRING_DATASOURCE_URL}
networks:
  - internal_network
```

```
networks:
  internal_network:
```

9) 셸 스크립트 작성 : deploy.sh

```
cd ~
sudo rm -rf mamdaero2

git clone https://lab.ssafy.com/s11-webmobile1-sub2/S11P12D103.git mamdaero2

echo "start build backend"
# build backend
cd ~/mamdaero2/backend
chmod 755 gradlew
cp ~/properties/application-secret.yml ~/mamdaero2/backend/src/main/resources
cp ~/properties/application.yml ~/mamdaero2/backend/src/main/resources
./gradlew clean build -x test
echo "end build backend"

cd ~/mamdaero2/frontend
cp ~/properties/.env ~/mamdaero2/frontend
sudo npm install
sudo npm run build

echo "end build frontend"

cd ~/mamdaero2
```

```
echo "docker compose up"
```

```
cd ~/deploy
```

```
sudo docker-compose down
```

```
sudo docker-compose up --build -d
```

3. 배포

- 1) 자동 배포 : Jenkins 빌드 성공시 자동 배포
- 2) 수동 배포 : ./deploy.sh

III. 시연 시나리오

시연 시나리오

1. 내담자 정보 확인하기

나의 상담 - 상담기록 확인

혜민 클릭

자가진단검사결과 클릭

일기클릭

2. 화상상담

내담자로 로그인 후 나의 상담 클릭 후 1:1 화상채팅을 클릭한다
상담사와 일치하는 상담 ID 에서 연결을 누르고 서로 연결이 되면
상담사가 기다리는 중이라는 말이 사라지고 상담사가 보인다.

연결이 되면 채팅도 가능하게 된다.

상담사: 우울증으로 예약을 하셨는데, 어떤 일이 있으신가요?

내담자: 저는 요즘 집에만 있으면 너무 우울해서 잠만 자게 됩니다...

3. 상담완료 체크 -> 메인 화면으로 가짐

화상상담을 하는 페이지 내에 상담완료 버튼 클릭

4. 상담 기록 확인

나의 상담 - 상담기록 확인

혜민 클릭

5. 상담 보고서 작성

상담 기록 보기에서 보고서 작성을 클릭한다.