

# 포팅 매뉴얼

## 포팅 매뉴얼

### 1. 개발 환경

#### 1) 프로젝트 기술 스택

##### Frontend

- Visual Studio Code(IDE): 1.97.2
- HTML5, CSS3, Javascript(ES2016)
- React: 19.0.0
- TypeScript: 5.7.2
- Tailwind CSS: 4.0.13
- Vite: 6.2.0
- NodeJS: 22.14.0
- 주요 라이브러리:
  - tanstack/react-query: 5.69.0
  - zustand: 5.0.3
  - axios: 1.8.4
  - react-router-dom: 7.3.0
  - sentry/react: 9.10.1

##### Backend

- IntelliJ(IDE): 2023.3.8
- JVM OpenJDK: 17
- Spring Boot: 3.4.3
- Gradle: 8.12.1

- 주요 라이브러리:
  - JWT: 0.12.3
  - Spring Security + OAuth2
  - Spring Data JPA
  - Spring Data Redis

## CI/CD

- AWS EC2 (개발 환경)
- GCP VM (운영 환경)
- Ubuntu: 22.04 LTS
- Docker: 27.5.1
- Jenkins: 2.492.1
- AWS S3
- Docker Hub

## 기타 서비스

- MSW (개발용 API 모킹)
- SonarQube (코드 품질 검사)
- Grafana & Prometheus (모니터링)
- Sentry (오류 모니터링)

## 2. 환경변수 설정

### 1) Backend: .env

```
MYSQL_HOST=mysql
MYSQL_PORT=3306
MYSQL_DATABASE=pic_score_db
MYSQL_USER=picscore_user
MYSQL_PASSWORD=mysecretpassword

MYSQL_ROOT_PASSWORD=rootpassword
REDIS_PASSWORD=mysecretpassword
```

REDIS\_HOST=redis  
REDIS\_PORT=6379

GOOGLE\_APP\_ID=458650965517-kbjhc62nc2obc2j0rre5a7f55umuae3r.ap  
ps.googleusercontent.com  
GOOGLE\_REDIRECT\_URL=https://picscore.net/login/oauth2/code/google  
GOOGLE\_BASE\_URL=https://accounts.google.com  
GOOGLE\_SECRET=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

KAKAO\_APP\_ID=6d909b2d412c05e14575fb26a4c6f12b  
KAKAO\_REDIRECT\_URL=https://picscore.net/login/oauth2/code/kakao

LOGIN\_SUCCESS=https://picscore.net?loginSuccess=true  
FIRST\_USER=https://picscore.net/agreement

JWT\_SECRET\_KEY=secure\_key\_with\_minimum\_32\_characters\_length!  
JWT\_ACCESS\_EXP=600000  
JWT\_REFRESH\_EXP=86400000  
JWT\_REDIS\_EXP=86400

AWS\_ACCESS\_KEY=AKIARO7555E366ZAWTNT  
AWS\_SECRET\_KEY=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  
AWS\_S3\_BUCKET=picscore-s3

GOOGLE\_CLOUD\_VISION\_API\_KEY=AlzaSyDluXqvUzBRQe4hOfODxavl8ML  
AIDW2ur0  
AZURE\_COMPUTER\_VISION\_KEY=FzJP7P9AkOYKLLbb3PL1aVLX66PDEufr  
49SbMxxRmD7OZGB2wXJsJQQJ99BCACNns7RXJ3w3AAAFACOGILsq  
AZURE\_ENDPOINT=https://kimsunjin.cognitiveservices.azure.com/

OPENAI\_API\_KEY=sk-proj-MlvGpQCVXu9aH6\_v-0XKupTc-ls1HYWm8D9RX  
UIXqeCuRBRQ1SklJCpQ7yYltfwc9TQK6zrSFbT3BIbkFJhFzQURrsD0lhdGbr  
F6-zjV0cl7fLWZ2z48C4TXtnKlrQKm3Xo8wf9OW2tz7ILWS19t3xE-H5gA

SENTRY\_DSN\_KEY=https://ddad7224c5340a0e840748574bf999fc@o450  
9077080834048.ingest.us.sentry.io/4509077096169472  
SENTRY\_ENV=production

```
SENTRY_NAME=picscore.net
```

```
AI_SERVER=http://15.164.216.52:8000
```

## 2) Frontend: .env.front

```
VITE_BASE_URL=https://picscore.net
```

## 3) MySQL: 컨테이너 환경변수 (docker-compose.yml에 통합됨)

```
MYSQL_ROOT_PASSWORD=rootpassword  
MYSQL_DATABASE=pic_score_db  
MYSQL_USER=picscore_user  
MYSQL_PASSWORD=mysecretpassword
```

## 4) Redis: 컨테이너 환경변수 (docker-compose.yml에 통합됨)

```
REDIS_PASSWORD=mysecretpassword
```

# 3. 프로젝트에서 사용하는 외부 서비스

## 1) 소셜 로그인

- Google OAuth2
- Kakao OAuth2

## 2) 클라우드 서비스

- AWS S3 (이미지 저장)
- AWS EC2 (개발 환경)
- GCP VM (운영 환경)

## 3) AI 서비스

- Azure Computer Vision
- OpenAI API

## 4) 모니터링

- Prometheus (메트릭 수집)
- Grafana (대시보드)
- Sentry (에러 트래킹)

## 4. 빌드 및 배포 방법

### 1) 소스 코드 클론

```
git clone https://lab.ssafy.com/s12-ai-image-sub1/S12P21B104.git
cd S12P21B104
```

### 2) 환경 변수 파일 설정

프로젝트 루트 디렉토리에 다음 파일들을 생성하고 환경에 맞게 값을 설정합니다:

- `.env` (백엔드 환경변수)
- `.env.front` (프론트엔드 환경변수)
- `redis.conf` (Redis 설정 파일 - 비밀번호 설정 포함)

### 3) Docker Compose로 배포

프로젝트의 루트 디렉토리에서 다음 명령어로 전체 애플리케이션을 배포합니다:

```
docker compose up -d
```

위 명령어는 다음 컨테이너들을 구동합니다:

- frontend: React 프론트엔드 애플리케이션
- backend: Spring Boot 백엔드 애플리케이션
- mysql: 데이터베이스
- redis: 캐시 및 세션 스토리지
- proxy: Nginx 웹 서버 (리버스 프록시)
- prometheus: 모니터링 데이터 수집
- grafana: 모니터링 대시보드
- node-exporter: 시스템 메트릭 수집

- cadvisor: 컨테이너 메트릭 수집

## 4) SSL 인증서 설정 (운영 환경)

운영 환경에서는 Let's Encrypt 인증서를 사용합니다. 인증서를 발급받고 적절한 위치에 저장한 후 nginx 설정을 수정해야 합니다.

```
# Certbot 설치
sudo apt-get update
sudo apt-get install certbot

# 인증서 발급
sudo certbot certonly --standalone -d your-domain.com

# 인증서 위치 확인
ls -la /etc/letsencrypt/live/your-domain.com/
```

## 5) 배포 자동화 (Jenkins)

Jenkins를 사용하여 CI/CD 파이프라인을 구축했습니다. Jenkins 설정은 프로젝트 루트의 `Jenkinsfile`에 정의되어 있습니다.

다음과 같은 파이프라인이 구성되어 있습니다:

1. 소스 코드 체크아웃
2. 환경 변수 설정
3. SonarQube 코드 품질 분석 (develop 브랜치)
4. Docker 이미지 빌드
5. Docker Hub에 이미지 푸시
6. 환경별 배포 (develop → EC2, master → GCP)

## 5. 시연 시나리오

### 1) 웹 접속

- <https://picscore.net> 접속

### 2) 사용자 인증

1. 화면 우측 상단의 로그인 버튼 클릭

2. Google 또는 Kakao 계정으로 소셜 로그인
3. 최초 로그인 시 약관 동의 및 프로필 설정

### 3) 사진 평가

1. 홈 화면에서 "Upload" 버튼 클릭
2. 사진 업로드 또는 촬영
3. AI의 사진 평가
4. 피드백 및 태그 작성 후 게시물 업로드 여부 결정
5. 업로드 완료 후 피드에서 확인

### 4) 게이미피케이션 액티비티

#### 타임 어택

1. 액티비티 메뉴에서 "타임 어택" 선택
2. 시스템이 제시한 제시어 확인 (예: "컵", "키보드", "나무" 등)
3. 제한 시간 내에 제시어에 맞는 사진 촬영 및 업로드
4. 시간 내 성공 시 포인트 획득
5. 랭킹 보드에서 자신의 순위 확인

#### 아레나

1. 액티비티 메뉴에서 "아레나" 선택
2. 제시된 여러 장의 사진 확인
3. AI가 평가한 점수를 예측하여 사진을 순서대로 배치
4. 실제 AI 평가 점수와 비교하여 정확도에 따른 점수 획득
5. 랭킹 보드에서 다른 사용자들과의 경쟁 순위 확인

### 5) 업적 시스템

1. 프로필 페이지에서 "업적" 탭 선택
2. 획득한 업적 뱃지 확인
  - 첫 팔로워
  - 인기 크리에이터

- 첫 사진 평가
- 평가 마스터
- 첫 게시글
- 콘텐츠 크리에이터
- 첫 타임어택 90점
- 타임어택 중독자
- 고품질 사진 작가
- 타임어택 챔피언
- 인기 콘텐츠
- 업적 마스터

## 6) 사용자 피드

1. 내 프로필 확인
2. 업로드한 사진 목록 확인
3. 비공개 사진 목록 확인
4. 액티비티 참여 이력 및 획득 포인트 확인

## 7) 탐색

1. 탐색 탭에서 인기 사진 확인
2. 최신순, 좋아요순, 점수순 사진 정렬
3. 사용자 및 게시물의 태그 검색 가능

## 8) 소셜 기능

1. 다른 사용자 프로필 방문
2. 팔로우/팔로잉 확인
3. 랭킹 보드에서 친구 및 전체 순위 확인

이 시연 시나리오는 PicScore 서비스의 주요 기능을 포함하고 있습니다. 사용자는 사진을 업로드하고 AI에게 사진을 평가받는 기본 기능뿐만 아니라, 타임 어택과 아레나 같은 게이미피케이션 요소를 통해 더욱 몰입감 있게 서비스를 이용할 수 있습니다. 또한 업적 시스템을 통해 사용자의 지속적인 참여를 유도하고 성취감을 제공합니다.



