



삼성SW청년아카데미 대전캠퍼스 7기

B205 | 최진호 김정윤 박서은 이길상 진윤아

1. 배포 상세

- git 저장소 clone

```
git clone https://lab.ssafy.com/s07-bigdata-recom-sub2/S07P22B205.git
```

- Docker 및 docker-compose 설치

```
# docker 설치 스크립트를 wget으로 가져와서 실행
wget -qO- https://get.docker.com/ | sh
# 설치 확인
docker version

# docker compose 확인
docker compose version
# 설치되어 있지 않다면 설치 진행
apt-get install docker-compose-plugin
```

- 빌드 파일 작성

- nginx 도커파일

```
FROM node:16.17.0 as builder
# 작업 폴더로 소스 파일 복사
COPY {git 폴더}/front/sharkshark /home/react
WORKDIR /home/react
# node 패키지 설치 후 빌드
RUN npm install
RUN npm run build

FROM nginx # nginx 설정 복사
COPY {nginx.conf 위치} /etc/nginx
# 빌드 파일 복사
COPY --from=builder /home/react/build /home/build
# 포트 개방
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- fastapi 도커파일

```
FROM python:3.9
# 작업 폴더로 실행 폴더 복사
WORKDIR /code
COPY {git 폴더}/back/pythonProject /code
# 파이썬 패키지 설치 후 실행
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

- nginx.conf

```
user nginx;
worker_processes auto;
events {
    worker_connections 1024;
}
http{
    include mime.types;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    server {
        // 포트 지정
        listen 80;
        listen [::]:80;

        // 프론트 빌드파일 경로설정
        location / {
            root /home/build;
            index index.html index.htm;
            try_files $uri $uri/ /index.html;
        }
        // 백엔드 api 요청 포워딩
        location /api/{
            proxy_pass http://172.17.0.1:8000/;
        }
    }
}
```

- docker-compose.yml

```
user nginx;
worker_processes auto;
events {
    worker_connections 1024;
}
http{
    include mime.types;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    server {
        // 포트 지정
        listen 80;
        listen [::]:80;

        // 프론트 빌드파일 경로설정
        location / {
            root /home/build;
            index index.html index.htm;
            try_files $uri $uri/ /index.html;
        }
        // 백엔드 api 요청 포워딩
        location /api/{
            proxy_pass http://172.17.0.1:8000/;
        }
    }
}
```

- docker-compose.yml

```
version: '3'
services:
  nginx:
    build:
      context: .
      dockerfile: {nginx dockerfile 이름}
    ports:
      - 80:80
  api:
    build:
      context: .
      dockerfile: {fastapi dockerfile 이름}
    ports:
      - 8000:8000
    extra_hosts:
      - "localhost:host-gateway"
```

- 컨테이너 실행

- mysql 이미지 실행

```
# mysql 이미지 가져오기
docker pull mysql

# 컨테이너 실행
docker run --name mysql -e MYSQL_ROOT_PASSWORD={비밀번호} -d -p 3306:3306 mysql
```

- 3306 포트로 MySQL 접속하여 b205 스키마 생성

- docker-compose 실행

```
docker compose up -d --build
# 혹은
docker-compose up -d --build
```

- 작동 확인

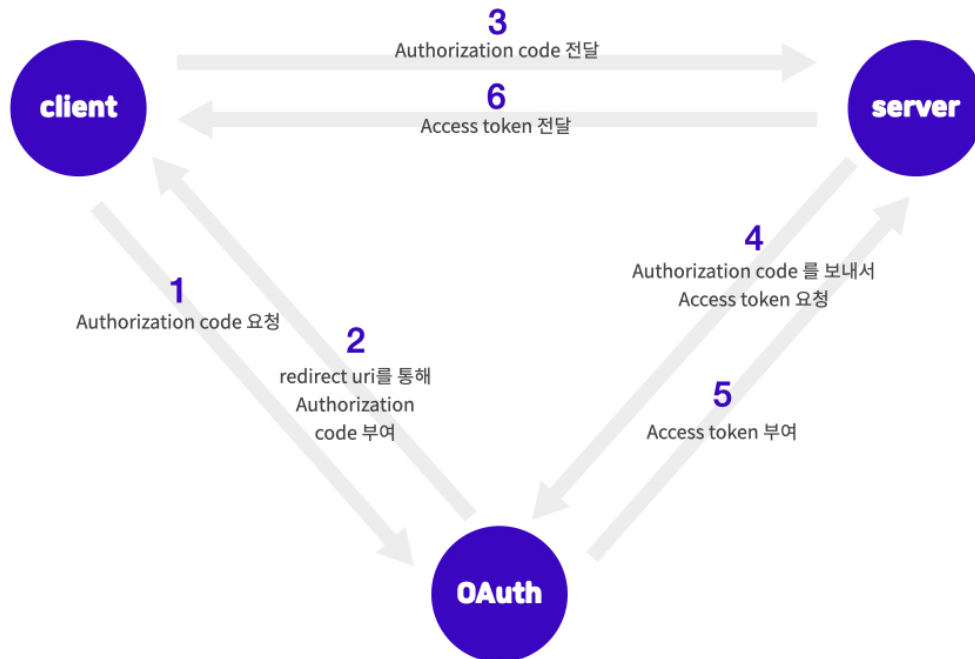
- 실행중인 컨테이너 조회

```
docker compose up -d --build
# 혹은
docker-compose up -d --build
```

- mySQL 접속하여 DB 덤프 파일 (/exec/sharkshark_dp_dump.zip) 실행

```
docker compose up -d --build
# 혹은
docker-compose up -d --build
```

2. Github OAuth 연동




CODE STATES

- Developer applications (github.com) 로 이동.
- OAuth Apps에 사이트 ip 또는 도메인 등록.

General
Optional features
Advanced

SSAFY


 Nelfunction owns this application.

You can list your application in the [GitHub Marketplace](#) so that other users can discover it.

8 users

Client ID
9539ff1ae93c2ccb932b

Client secrets

 *****782ecb16
Added 10 days ago by Nelfunction
Last used within the last week
You cannot delete the only client secret. Generate a new client secret first.

Transfer ownership
List this application in the Marketplace
Revoke all user tokens
Generate a new client secret
Delete

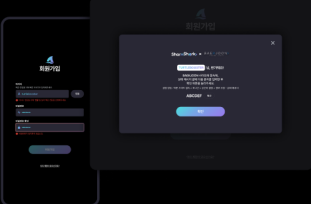
- client_id 와 client_secrets 발급.
- front 및 back 소스 파일의 키 대체하여 사용.

3. 주요 기능

서비스 소개

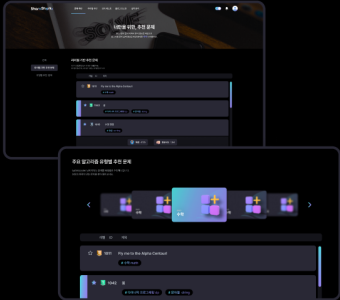
주요 기능

01



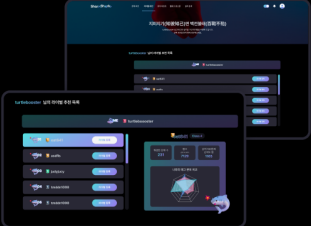
회원 가입 시 백준 연동 인증

02

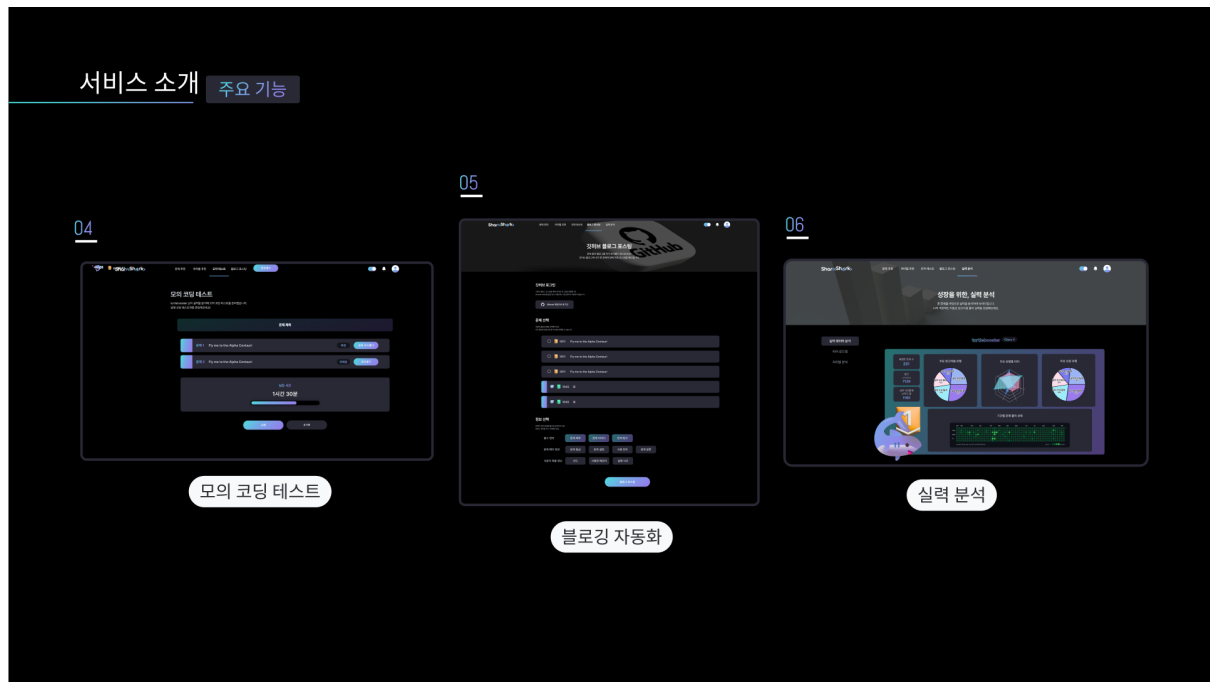


문제 추천

03



라이벌 추천



1. 회원 가입 시, 백준 연동 인증

2. 문제 추천

- 추천 문제 : user와 비슷한 실력을 가진 사용자 기반하여 추천
- 사용자 기반 추천 문제 : user의 라이벌 기반 추천 문제, 라이벌이 푼 문제 중 user가 풀지 않은 문제 추천
- 유형별 추천 문제 : 주요 알고리즘 유형별(총 8유형) 추천 문제
 - 주요 알고리즘 유형 : math, implementation, greedy, string, data structures, graphs, dp, bruteforce

3. 라이벌 추천

- 라이벌 추천 목록 : user와 비슷하거나 조금 상위에 있는 라이벌을 추천
 - 라이벌 등록 전, 추천 라이벌과 user의 실력 분석 기능
- 라이벌 관리 : 라이벌 등록/해지

4. 모의 코딩 테스트

- 문제 : 추천 문제 리스트 중에서 추천
- 코딩 테스트 시간 선정 기준 : 추천된 문제들의 level의 평균(문제 난이도)을 고려
- 문제 제출 : BOJ에서 문제 제출 여부 확인
- 테스트 종료 후, 제출한 문제들에 대한 실력 분석 제공

5. 블로깅 자동화

- 블로그 계정 설정 : Github 계정과 연결, 업로드할 Github repository 선택

- Github 저장소에 markdown 파일 업로드

6. 실력 분석

- 알고리즘 실력 분석
- 티어 로드맵
- 유사 사용자 분석