포팅 메뉴얼

1.1 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전

- JVM: OpenJDK 17
- 웹서버: Nginx 1.24
- WAS: Spring Boot 3.3.6 (Embedded Tomcat 사용)
- IDE: IntelliJ IDEA 2023.3 / VS Code
- Fluttter: 3.7.0

1.2 환경변수

프로젝트에서 사용되는 주요 환경변수는 다음과 같습니다:

데이터베이스 관련 환경변수

- `MYSQL_URL`: MySQL 데이터베이스 연결 URL
- `MYSQL_USERNAME`: MySQL 사용자 이름
- `MYSQL_PASSWORD`: MySQL 사용자 비밀번호
- `MONGODB_URI`: MongoDB 연결 URI

캐시 및 메시지 큐 관련 환경변수

- `REDIS_HOST`: Redis 서버 호스트 주소
- `REDIS_PASSWORD`: Redis 접속 비밀번호
- `KAFKA_HOST`: Kafka 부트스트랩 서버 주소

블록체인 관련 환경변수

- `HOLESKY_ENDPOINT`: Holesky 테스트넷 엔드포인트
- `SEPOLIA_WEBSOCKET_ENDPOINT`: Sepolia 테스트넷 웹소켓 엔드포인트
- `SEPOLIA_NODE_ENDPOINT`: Sepolia 테스트넷 노드 엔드포인트
- `INFURA_API_KEY`: Infura API 키

- `INFURA_API_KEY_SECRET`: Infura API 키 시크릿
- `STREAMING_AGGREGATION_CONTRACT_ADDRESS`: 스트리밍 집계 컨트랙트 주소
- `SUBSCRIPTION_CONTRACT_ADDRESS`: 구독 컨트랙트 주소
- `OWNER_ADDRESS`: 소유자 계정 주소
- `OWNER_PRIVATE_KEY`: 소유자 계정 개인키

파일 저장소 관련 환경변수

- `AWS_ACCESS_KEY`: AWS S3 접근 키
- `AWS_SECRET_KEY`: AWS S3 비밀 키
- `AWS_S3_BUCKET_NAME`: AWS S3 버킷 이름
- `IPFS_API_URL`: IPFS API URL
- `IPFS_LOCAL_NODE_URL`: IPFS 로컬 노드 URL
- `IPFS_USE_LOCAL_CACHE`: IPFS 로컬 캐시 사용 여부
- `PINATA_ENDPOINT`: Pinata 엔드포인트
- `PINATA_JWT`: Pinata JWT 토큰
- `PINATA_GATEWAY`: Pinata 게이트웨이

인증 관련 환경변수

- `JWT_SECRET_KEY`: JWT 토큰 암호화 키
- `GOOGLE_CLIENT_ID`: Google OAuth 클라이언트 ID
- `API_KEY_NAME`: API 키 이름
- `API_KEY_SECRET`: API 키 비밀

1.3 배포 특이사항

Docker 배포

- 제공된 Dockerfile을 사용하여 Docker 컨테이너로 배포합니다.
- 시간대는 'Asia/Seoul'로 설정되어 있습니다.
- 로그 디렉토리는 `/app/logs`에 생성됩니다.

- 서버는 8080 포트를 사용합니다.

백엔드 배포 과정

1. 프로젝트 클론

```
bash
[ ]

git clone https://lab.ssafy.com/s12-blochain-transaction-sub1/S12P21C205.git
```

2. MySQL 설치 및 설정

```
bash
[ ]

apt update
apt install mysql-server
sudo mysql -u root -p
create database;
```

   - 기본 포트는 3306을 사용하며, 필요시 변경 가능합니다.
   - MySQL 사용자 생성 및 root 비밀번호 설정이 필요합니다.

3. 백엔드 빌드 및 배포
   - backend 디렉토리 루트 경로에서 다음 명령어 실행:

```
bash
[ ]

# Java 11 기반 Docker 이미지 사용
docker build -t backend-app .
docker run -p 8080:8080 --name backend-container backend-app
```

4. Nginx 설정 및 SSL 인증서 적용
   - Nginx 웹 서버를 통한 리버스 프록시 설정이 필요합니다.
   - SSL 인증서를 발급받아 HTTPS 통신을 적용합니다.

5. APK 앱 빌드
    - FLUTTER, ANDROID SDK가 필요합니다.
    - flutter doctor를 통해 필수 요소들을 설치합니다.
    - 이후 flutter build apk —release 명령어 수행 시 빌드와 패키징이 진행됩니다.

블록체인 관련 특이사항

- 코인 지갑 설치가 필요합니다.
- IPFS와 스마트 컨트랙트가 사용되므로 관련 환경 구성이 필요합니다.
- 테스트넷(Holesky, Sepolia)에 배포된 컨트랙트와 연동됩니다.

2. 외부 서비스

블록체인 서비스

- **Infura**: 이더리움 노드 접근을 위한 API 서비스
    - 필요 정보: API 키, API 시크릿
    - 용도: 스마트 컨트랙트 상호작용 및 트랜잭션 전송
- **IPFS(InterPlanetary File System)**: 분산 파일 저장 시스템
    - 로컬 노드 또는 원격 API 사용 가능
    - 필요 정보: IPFS API URL, 로컬 노드 URL
- **Pinata**: IPFS 파일 핀 서비스
    - 필요 정보: 엔드포인트, JWT 토큰, 게이트웨이
    - 용도: IPFS에 업로드된 파일의 지속적인 가용성 보장

클라우드 스토리지

- **AWS S3**: 파일 저장소
    - 필요 정보: 접근 키, 비밀 키, 버킷 이름
    - 용도: 미디어 파일 및 기타 데이터 저장

메시지 큐

- **Kafka**: 이벤트 스트리밍 플랫폼
  - 필요 정보: 부트스트랩 서버 주소
  - 용도: 스트리밍 이벤트 처리

인증 서비스

- **Google OAuth**: 소셜 로그인
  - 필요 정보: 클라이언트 ID
  - 용도: 사용자 인증

데이터베이스 서비스

- **MySQL**: 관계형 데이터베이스
  - 필요 정보: URL, 사용자 이름, 비밀번호
  - 용도: 주요 데이터 저장
- **MongoDB**: NoSQL 데이터베이스
  - 필요 정보: URI
  - 용도: 비정형 데이터 저장
- **Redis**: 인메모리 데이터 저장소
  - 필요 정보: 호스트 주소, 포트, 비밀번호
  - 용도: 캐싱 및 세션 관리

4. 시연 시나리오

1. 시작 화면

1. 로그인 (아티스트랑 리스너 둘 다 되어있는 계정)

- 리카스

2. 메인 화면

1. 메인화면 보여주기

- 최신앨범, 인기 앨범, 인기 플레이리스트, HOT 20를 스와이프 하면서 소개
- 장르별로 클릭하면서

2. 현재 차트는 이렇게 집계되고 있다.
3. 검색 화면으로 이동

3. 검색 화면

1. 장르별 카테고리 소개 후 원하는 장르 카테고리로 들어가기
2. 장르에서 원하는 앨범 클릭해서 앨범 상세 보여주기
3. 앨범 상세를 소개하면서 등록되어 있는 트랙 클릭해서 트랙 상세로 들어가기
    - 첫번째 핵심 기능 - 트랙 상세 페이지의 스트리밍 내역
        - 스트리밍 횟수랑 스트리밍 내역 보여주기
        - 현재 저희는 온체인 형식으로 스트리밍 기록을 저장하고 있다.
4. 아래 내비게이션 바를 통해 검색 페이지로 다시 가기
5. 현재 구독한 “강지철철” 검색 후 들어가기
6. 강지철철에 채널 보여주기

4. 구독한 아티스트 채널 소개

1. 강지철철 아티스트의 채널을 소개
2. 팬톡, 공지 사항, 공개된 플레이리스트 등등
3. 아티스트 채널에서 나와서 마이페이지로 이동

5. 마이페이지

1. 나의 구독, 구독 내역, 아티스트 대시 보드, 정산 내역을 보여줌
2. 아티스트 대시 보드 보여주기
3. 그 다음 나의 구독 내역 보여주기 - 아티스트 구독이 되어있음을 보여주기
4. 앨범 업로드 하기
5. 채널 이동 or 마이페이지 이동이 뜨면 → 채널 이동 선택해서 나의 채널로

6. 나의 채널

1. 나의 앨범 확인
6. 나의 앨범 조회 후 트랙 상세로 들어가기
7. 트랙 상세 페이지에서 나의 노래를 재생
8. 재생 목록 탭에서 재생되고 있는지 확인
9. 재생 목록에서 노래를 플레이리스트에 담기 (플레이리스트를 새로 생성 - 공개 형태로)
10. 재생 탭에서 플레이리스트 눌러서 선택한 트랙이 들어왔는지 확인 후
11. 탭을 내리고 나의 채널에서 공개된 플레이리스트가 나오는지 확인