

이젠 아카데미(성남 분당)

HeartBeat



It makes your Heartbeat,
당신의 심장을 뛰게 하는 음악 커뮤니티 사이트

2024.12.20

김혜인, 권나영, 김홍현, 윤보현, 이영훈

목차

1. 프로젝트 개요
2. 팀 구성 및 역할
3. 데이터베이스 모델링 및 구축
4. 프로젝트 구현 내용
5. 주요 기능 시연
6. 수정 및 발전 방향

01 프로젝트 개요

음악 스트리밍과 팬 커뮤니티 기능을 결합하여 다양한 사용자 경험을 제공하는 웹 애플리케이션



- 일간, 주간, 실시간 음악 차트 제공
- 사용자가 해시태그를 선택해 사용자에게 맞는 노래 추천



- 멤버십을 통해 내가 선택한 아티스트와 소통
- 같은 팬덤의 다른 유저들과 다양한 소통



- 어드민 모드에서 회원, 작성글, 멤버십 등 손쉽게 사이트 관리 및 유지

01 프로젝트 개요 - 기술적 목표

다양한 외부 API와의 통합을 통해 실제 서비스를 구현을 학습
데이터 처리, 사용자 경험 개선, 보안 등을 학습하고 강화



01 프로젝트 개요 - 개발 툴 및 기술 스택

개발 툴

FE



BE

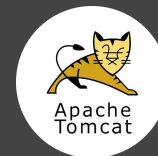


Co-Work



기술 스택

라이브러리



DB

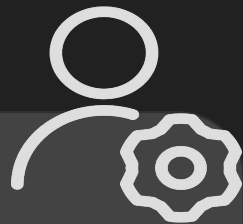


사용자



- 1. 로그인**
 - └ 이메일 찾기, 비밀번호 찾기, 비밀번호 찾기 시 가입 할 때 입력한 이메일로 새로운 비밀번호 발송
- 2. 회원가입**
 - └ 이메일, 닉네임, 핸드폰 번호 중복 확인, 프로필 사진 업로드
- 3. 음악 차트**
 - └ 회원 가입만 하면 사용할 수 있는 서비스. 일간, 주간, 실시간 인기 순위의 노래 확인, 음악 듣기 가능
- 4. 플레이리스트**
 - └ 멤버십 레벨 1 이상의 사용자만 이용 가능한 서비스. 해시태그를 선택하면 해당 해시태그와 연관된 음악을 추천
- 5. 커뮤니티**
 - └ 멤버십 레벨 2 이상의 사용자만 이용 가능한 서비스. 사용자가 구매한 멤버십의 아티스트별로 접근할 수 있는 커뮤니티 게시판이 다르게 제공
- 6. 멤버십**
 - └ 레벨 1과 레벨 2 두 가지 타입의 멤버십을 제공. 나이스페이, 카카오페이 결제 시스템을 제공
- 7. 공지 및 문의**
 - └ 공지사항을 확인, 문의글 작성
- 8. 마이페이지**
 - └ 비밀번호, 닉네임, 프로필 사진, 구매한 멤버십 변경. 사용자가 작성한 게시글을 확인 및 삭제 가능

관리자



1. 요약 정보

└ 시스템의 전체적인 사이트의 정보를 실시간으로 모니터링할 수 있는 대시보드 기능을 제공

2. 회원 관리

└ 일반 사용자와 직원을 추가, 수정, 삭제 가능

3. 게시글 관리

└ 사용자가 작성한 게시글 삭제 및 관리 가능

02 팀 구성 및 역할

02 팀 구성 및 역할

이름	담당 업무
공통	<ol style="list-style-type: none">1. 디자인 및 기능 기획 : 전체적인 UI/UX 디자인과 프로젝트의 주요 기능에 대한 기획2. DB 구축 : MariaDB를 사용하여 효율적이고 체계적인 데이터베이스 설계 및 구축
김혜인 (팀장)	<ol style="list-style-type: none">1. 전체적인 프론트엔드 개발 : 사용자 UI의 전체적인 디자인 및 개발, 레이아웃과 기능을 설계, 반응형 웹 디자인을 적용2. Spotify API 세팅 : Spotify API를 초기 세팅3. 플레이리스트 구현 : 해시태그 기반 음악 추천 및 사용자 맞춤형 플레이리스트 기능을 구현
권나영	<ol style="list-style-type: none">1. 로그인 및 회원가입 기능 구현 : 로그인, 회원가입, 이메일, 닉네임, 핸드폰 번호 중복 확인, 비밀번호 찾기 등을 구현2. 마이페이지 기능 구현 : 사용자가 비밀번호, 닉네임, 프로필 사진 등을 수정할 수 있는 마이페이지 기능을 구현3. 유튜브 API를 이용한 재생 기능 구현 : 유튜브 API를 활용하여 음악을 스트리밍하고, 사용자가 실시간 인기 순위에 따른 노래를 들을 수 있도록 구현4. 차트 기능 구현 : Spotify에서 제공하는 일간, 주간, 실시간 인기 순위에 맞춰 사용자가 최신 음악을 스트리밍할 수 있는 차트 기능을 개발

02 팀 구성 및 역할

이름	담당 업무
김홍현	<ol style="list-style-type: none">1. 초기 DB 구축 및 세팅 : MariaDB를 사용하여 초기 데이터베이스 설계 및 구축, 테이블 구조 및 관계를 설정2. 커뮤니티 기능 구현 : 게시글 작성, 수정, 삭제 기능을 개발3. 쿼리 구현 : 데이터의 조회 및 관리를 위한 SQL 쿼리 작성4. 멤버십 결제 구현 : 카카오페이 API 및 나이스페이먼트 API를 사용하여 유료 멤버십 결제 시스템을 구현5. 공지 및 문의 페이지 구현 : 관리자 공지사항 및 문의글을 작성, 확인 할 수 있는 페이지를 구현
윤보현	<ol style="list-style-type: none">1. 관리자 페이지 구현 : 관리자 시스템을 구축, 회원 추가, 삭제, 게시글 삭제 기능 구현2. 쿼리 구현 : 관리자 기능에 필요한 SQL 쿼리 작성3. 초기 DB 구축 세팅 : DB 설계, DB 구조와 관계 설정을 지원, DB 테이블을 생성
이영훈	<ol style="list-style-type: none">1. 커뮤니티 페이지 세부 기능 구현 : 게시글 작성, 수정, 삭제 기능을 포함한 커뮤니티 페이지의 세부적인 부분을 구현2. 파일 업로드 기능 구현 : 파일 업로드, 수정, 삭제 기능 구현

03 데이터베이스 모델링 및 구축

03 데이터베이스 모델링 및 구축



해시태그		hashtag_tbl	
해시태그 아이디	hashtag_id	INT	NOT NULL
음악 아이디	music_id	INT	NOT NULL
아티스트 아이디	artist_id	INT	NOT NULL
해시태그 이름	hashtag_name	VARCHAR(50)	NOT NULL

음악리스트		music_tbl	
음악 아이디	music_id	INT	NOT NULL
아티스트 아이디	artist_id	INT	NOT NULL
곡명	music_name	VARCHAR(45)	NOT NULL
트랙 이름	track	VARCHAR(45)	NOT NULL
재생시간	playtime	INT	NOT NULL

아티스트(artist)		artist_tbl	
아티스트 아이디	artist_id	INT	NOT NULL
아티스트 이름	art_name	VARCHAR(100)	NOT NULL
팬덤	fan	VARCHAR(45)	NOT NULL
이미지 주소	art_img	VARCHAR(255)	NOT NULL

구독		Subscription_tbl	
no	구독번호	int	NOT NULL
구독_등급	level2	TINYINT	NOT NULL
이메일	email	VARCHAR(50)	NOT NULL
아티스트 아이디	artist_id	INT	NOT NULL
구독_시작날짜	start_date	DATE	NOT NULL
구독_종료날짜	end_date	DATE	NOT NULL

구독가격		Cost_tbl	
구독_등급	level	TINYINT	NOT NULL
가격	cost	INT	NOT NULL

역할		role_tbl	
역할_아이디	role_id	INT	NOT NULL
역할 이름	role_name	VARCHAR(50)	NOT NULL

유저		user_tbl	
이메일	email	VARCHAR(50)	NOT NULL
아티스트 아이디	artist_id	INT	NOT NULL
비밀번호	pwd	VARCHAR(20)	NOT NULL
이름	name	VARCHAR(40)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
생년월일	birth	DATE	NOT NULL
휴대전화	phone	VARCHAR(13)	NOT NULL
프로필사진	profile_img	VARCHAR(100)	NULL
구독_등급	level	INT	NULL
생성일	reg_date	DATETIME	NOT NULL
수정일	up_date	DATETIME	NOT NULL

유저_역할(ur_)		user_role_tbl	
유저역할	ur_key	INT(AUTO_INCREMENT)	not null
이메일	email	VARCHAR(50)	NOT NULL
역할_아이디	role_id	INT	NOT NULL

공지 및 문의 게시물		notice_tbl	
공지_아이디	notice_id	INT	NOT NULL
이메일	email	VARCHAR(50)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
공지_제목	title	TEXT	NOT NULL
공지_내용	content	TEXT	NOT NULL
작성일	post_date	DATETIME	NOT NULL

게시물 좋아요		like_tbl	
좋아요	like_id	INT(AUTO_INCREMENT)	NOT NULL
이메일	email	VARCHAR(50)	NOT NULL
게시물_아이디	post_id	INT	NOT NULL

게시물(post)		post_tbl	
게시물_아이디	post_id	INT	NOT NULL
이메일	email	VARCHAR(50)	NOT NULL
아티스트_아이디	artist_id	INT	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
게시물_내용	content	TEXT	NOT NULL
게시물_이미지	post_img	VARCHAR(100)	NULL
작성일	post_date	DATETIME	NOT NULL

댓글(cm)		comment_tbl	
댓글_아이디	comment_id	INT	NOT NULL
이메일	email	VARCHAR(100)	NOT NULL
게시물_아이디	post_id	INT	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
댓글_내용	comment	TEXT	NOT NULL
작성일	comment_date	DATETIME	NOT NULL

공지 댓글		notice_comment_tbl	
댓글_아이디	notice_comment_id	INT	NOT NULL
게시물_아이디	notice_id	INT	NOT NULL
이메일	email	VARCHAR(100)	NOT NULL
이메일	email2	VARCHAR(50)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
댓글_내용	comment	TEXT	NOT NULL
작성일	comment_date	DATETIME	NOT NULL

1. 주요 목적

- 사용자와 음악 관련 데이터를 효율적으로 저장하고 관리
- 게시물 및 공지사항 관리
- 구독, 역할 및 해시태그 등의 기능을 지원

2. 다이어그램 개요

- 데이터 무결성을 유지
- 각 테이블 간 관계를 적절히 활용해 설계

3. 설계 특성

- 정규화를 통해 중복 데이터를 최소화
- 기본 키(PK)와 외래 키(FK)를 설정해 관계형 데이터베이스의 특성을 강화

03 데이터베이스 모델링 및 구축

구독		Subscription_tbl	
PK	no	구독번호	int NOT NULL
	구독_등급	level2	TINYINT NOT NULL
	이메일	email	VARCHAR(50) NOT NULL
	아티스트_아이디	artist_id	INT NOT NULL
	구독_시작날짜	start_date	DATE NOT NULL
	구독_종료날짜	end_date	DATE NOT NULL

구독가격		Cost_tbl	
PK	구독_등급	level	TINYINT NOT NULL
	가격	cost	INT NOT NULL

역할		role_tbl	
PK	역할_아이디	role_id	INT NOT NULL
	역할_이름	role_name	VARCHAR(50) NOT NULL

유저		user_tbl	
PK	이메일	email	VARCHAR(50) NOT NULL
	아티스트_아이디	artist_id	INT NOT NULL
	비밀번호	pwd	VARCHAR(20) NOT NULL
	이름	name	VARCHAR(40) NOT NULL
	닉네임	nickname	VARCHAR(20) / UK NOT NULL
	생년월일	birth	DATE NOT NULL
	휴대전화	phone	VARCHAR(13) NOT NULL
	프로필사진	profile_img	VARCHAR(100) NULL
	구독_등급	level	INT NULL
	생성일	reg_date	DATETIME NOT NULL
	수정일	up_date	DATETIME NOT NULL

유저_역할(ur_)		user_role_tbl	
PK	유저역할	ur_key	INT(AUTO_INCREMENT) not null
	이메일	email	VARCHAR(50) NOT NULL
	역할_아이디	role_id	INT NOT NULL

1. 사용자 관리

1) user_tbl

- 사용자 정보를 저장
- 이메일을 기본 키로 설정해
고유한 식별 가능

2) role_tbl & user_role_tbl

- 다중 역할을 관리

3) Subscription_tbl

- 구독 정보와 비용을 연결해
사용자 별 구독 상태를 추적

03 데이터베이스 모델링 및 구축

해시태그		hashtag_tbl		
해시태그_아이디	hashtag_id	INT	NOT NULL	
음악_아이디	music_id	INT	NOT NULL	
아티스트_아이디	artist_id	INT	NOT NULL	
해시태그_이름	hashtag_name	VARCHAR(50)	NOT NULL	

음악리스트		music_tbl		
음악_아이디	music_id	INT	NOT NULL	
아티스트_아이디	artist_id	INT	NOT NULL	
곡명	music_name	VARCHAR(45)	NOT NULL	
트랙_이름	track	VARCHAR(45)	NOT NULL	
재생시간	playtime	INT	NOT NULL	

아티스트(art_)		artist_tbl		
아티스트_아이디	artist_id	INT	NOT NULL	
아티스트_이름	art_name	VARCHAR(100)	NOT NULL	
팬덤	fan	VARCHAR(45)	NOT NULL	
이미지_주소	art_img	VARCHAR(255)	NOT NULL	

유저		user_tbl		
이메일	email	VARCHAR(50)	NOT NULL	
아티스트_아이디	artist_id	INT	NOT NULL	
비밀번호	pwd	VARCHAR(20)	NOT NULL	
이름	name	VARCHAR(40)	NOT NULL	
닉네임	nickname	VARCHAR(20) / UK	NOT NULL	
생년월일	birth	DATE	NOT NULL	
휴대전화	phone	VARCHAR(13)	NOT NULL	
프로필사진	profile_img	VARCHAR(100)	NULL	
구독_등급	level	INT	NULL	
생성일	reg_date	DATETIME	NOT NULL	
수정일	up_date	DATETIME	NOT NULL	

2. 음악 관리

1) music_tbl

- 음악의 고유 ID와 관련된 데이터 저장

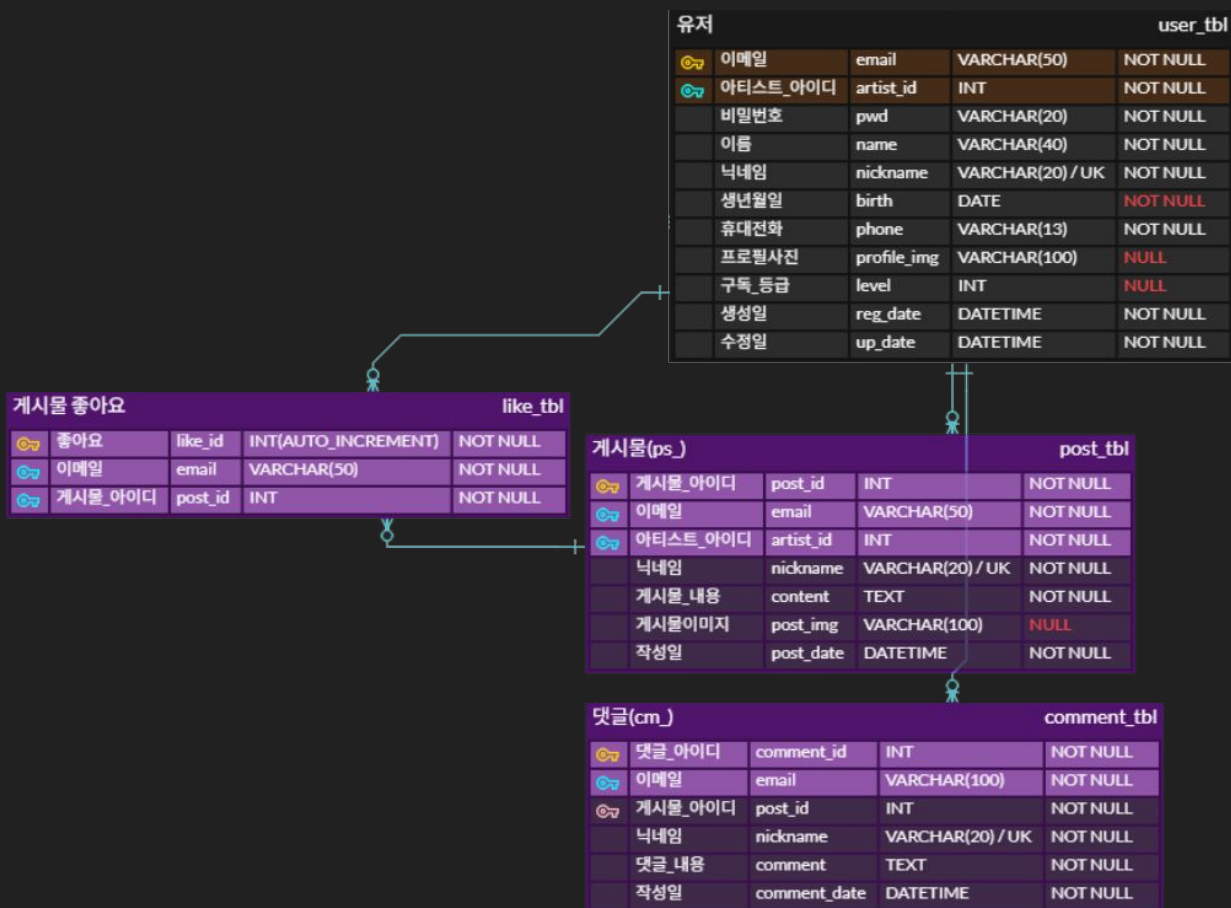
2) artist_tbl

- 아티스트의 정보 관리

3) hashtag_tbl

- 해시태그와 음악 매칭

03 데이터베이스 모델링 및 구축



3. 게시판 관리

- 1) post_tbl
- 사용자 게시물 데이터를 관리
- 작성자와 아티스트 ID를 연결
- 2) like_tbl
- 게시물에 대한 '좋아요'를 관리
- 3) comment_tbl
- 게시글의 댓글 저장

03 데이터베이스 모델링 및 구축

유저 user_tbl

이메일	email	VARCHAR(50)	NOT NULL
아티스트_아이디	artist_id	INT	NOT NULL
비밀번호	pwd	VARCHAR(20)	NOT NULL
이름	name	VARCHAR(40)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
생년월일	birth	DATE	NOT NULL
휴대전화	phone	VARCHAR(13)	NOT NULL
프로필사진	profile_img	VARCHAR(100)	NULL
구독_등급	level	INT	NULL
생성일	reg_date	DATETIME	NOT NULL
수정일	up_date	DATETIME	NOT NULL

공지 및 문의 게시물 notice_tbl

공지_아이디	notice_id	INT	NOT NULL
이메일	email	VARCHAR(50)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
공지_제목	title	TEXT	NOT NULL
공지_내용	content	TEXT	NOT NULL
작성일	post_date	DATETIME	NOT NULL

공지 댓글 notice_comment_tbl

댓글_아이디	notice_comment_id	INT	NOT NULL
게시물_아이디	notice_id	INT	NOT NULL
이메일	email	VARCHAR(100)	NOT NULL
이메일	email2	VARCHAR(50)	NOT NULL
닉네임	nickname	VARCHAR(20) / UK	NOT NULL
댓글_내용	comment	TEXT	NOT NULL
작성일	comment_date	DATETIME	NOT NULL

4. 공지사항 관리

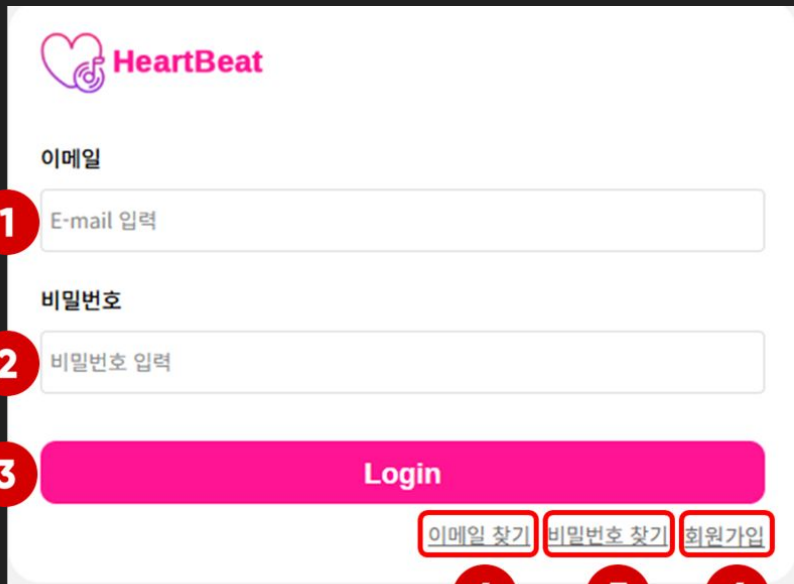
1) notice_tbl

- 공지사항, 문의사항 관리

2) notice_comment_tbl

- 공지사항 댓글 관리

04 프로젝트 구현내용



HeartBeat

이메일

1 E-mail 입력

비밀번호

2 비밀번호 입력

3 Login

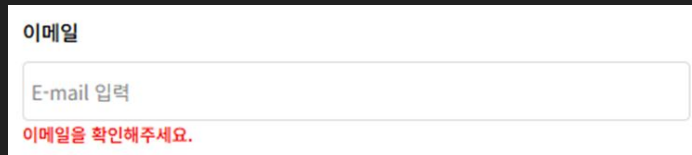
4 이메일 찾기

5 비밀번호 찾기

6 회원가입

1. 이메일 입력

- 이메일 잘못 입력했을 때



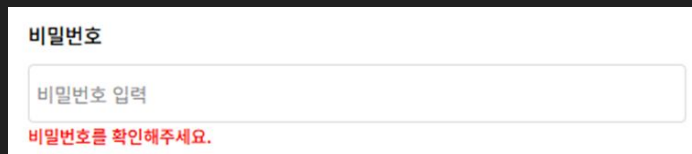
이메일

E-mail 입력

이메일을 확인해주세요.

2. 비밀번호 입력

- 비밀번호 잘못 입력했을 때



비밀번호

비밀번호 입력

비밀번호를 확인해주세요.

3. 로그인

4. 이메일 찾기

5. 비밀번호 찾기

6. 회원가입

4 이메일 찾기

- 사용자 입력 값 = DB 값
- 사용자 입력 값 \neq DB 값

이메일 찾기 ✕

이메일 찾기

김지훈님의 이메일은 user1@heartbeat.com입니다.

이메일 찾기 ✕

이메일 찾기


김지훈님의 이메일을 찾지 못했습니다. 다시 입력해주세요.

5 비밀번호 찾기

- 사용자의 이메일로 새 비밀번호 전송
javaMailSender 이용하여 Spring Framework에서 이메일 전송

비밀번호 찾기 ✕

비밀번호 찾기

☆ [Heartbeat] 새 비밀번호 안내입니다. 

보낸사람 harujongil999@gmail.com

받는사람

2024년 12월 9일 (월) 오후 4:39

새 비밀번호 : 9pvISYdh

- DB에 새 비밀번호로 저장

pwd ▲⁴

9pvISYdh

```
@Override
public int searchPwd(UserVO userVO) {
    int result = 0;
    String email = userVO.getEmail();
    UserVO uvo = userPersistenceImpl.searchPwd(userVO);

    if (uvo != null) {
        String newPassword = createNewPassword();
        result = userPersistenceImpl.updatePwd(email, newPassword);
        if (result > 0)
            sendNewPasswordByMail(email, newPassword, userVO);
    }
    return result;
}
```

1

3

```
private String createNewPassword() {
    StringBuffer stringBuffer = new StringBuffer();
    SecureRandom secureRandom = new SecureRandom();
    secureRandom.setSeed(new Date().getTime());

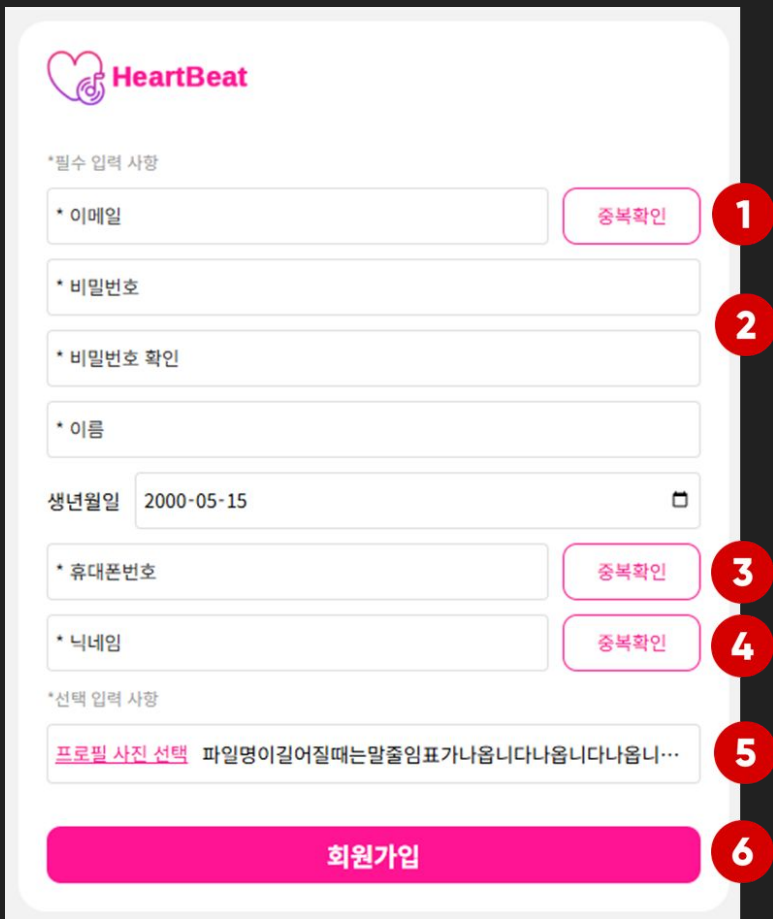
    int index = 0;
    int length = chars.length;
    for (int i = 0; i < 8; i++) {
        index = secureRandom.nextInt(length);
        if (index % 2 == 0)
            stringBuffer.append(String.valueOf(chars[index]).toUpperCase());
        else
            stringBuffer.append(String.valueOf(chars[index]).toLowerCase());
    }
    return stringBuffer.toString();
}
```

2

```
private void sendNewPasswordByMail(String toMailAddr, String newPassword, UserVO userVO) {
    final MimeMessagePreparator mimeMessagePreparator = new MimeMessagePreparator() {
        @Override
        public void prepare(MimeMessage mimeMessage) throws Exception {
            final MimeMessageHelper mimeMessageHelper = new MimeMessageHelper(mimeMessage, true, "UTF-8");
            mimeMessageHelper.setTo(userVO.getEmail());
            mimeMessageHelper.setSubject("[Heartbeat] 새 비밀번호 안내입니다.");
            mimeMessageHelper.setText("새 비밀번호 : " + newPassword, true);
        }
    };
    javaMailSenderImpl.send(mimeMessagePreparator);
}
```

4

1. 사용자의 정보 확인
2. 새로운 비밀번호 생성
3. 새로운 비밀번호 DB에 저장
4. 메일로 새로운 비밀번호 전송



HeartBeat

*필수 입력 사항

* 이메일 중복확인 1

* 비밀번호 2

* 비밀번호 확인

* 이름

생년월일 2000-05-15

* 휴대폰번호 중복확인 3

* 닉네임 중복확인 4

*선택 입력 사항

[프로필 사진 선택](#) 파일명이 길어질 때는 말줄임표가 나옵니다. 나옵니다. 나옵니다...

회원가입 6

1. 이메일 중복확인

- 이메일 입력 형식

```
const emailPattern = /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/;
```

2. 비밀번호 입력값이 서로 일치하면 DB에 암호화된 비밀번호 저장

- 비밀번호 입력 형식

```
const pwdVPattern = /^(?=.*[A-Z])(?=.*\d)(?=.*[\W_]).{8,16}$/;
```

3. 전화번호 중복 확인

4. 닉네임 중복확인

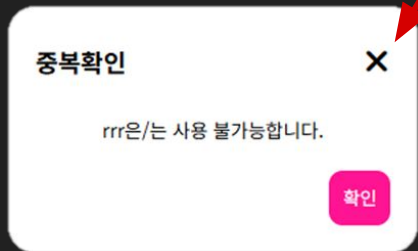
5. 프로필 이미지 넣기

6. 회원가입

04 - 2 회원가입

1 3 4 중복확인

- 입력값과 DB 불일치 시 사용 가능
- 중복확인을 하지 않으면 회원가입 불가



HeartBeat

*필수 입력 사항

• 이메일 중복확인 1

• 비밀번호 2

• 비밀번호 확인

• 이름

생년월일 2000-05-15

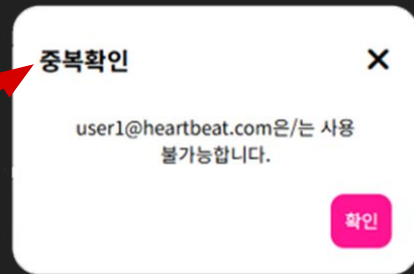
• 휴대폰번호 중복확인

• 닉네임 중복확인

*선택 입력 사항

프로필 사진 선택 파일명이 길어질때는 말줄임표가나옴니다나옴니다나옴니...

회원가입 6



2 비밀번호 암호화

- Spring Security에서 제공하는 BCryptPasswordEncoder 사용하여 암호화

```
pwd  
$2a$10$BZurYWgpw8khlsXeJ6NYb
```

1 Ready to play a Music

2 차트

오늘 Top 10

- 1 Who
Jimin
- 3 HOME SWEET HOME (feat. TAEYANG, DAESUNG)
G-DRAGON
- 5 Seven (feat. Latto) (Explicit Ver.)
Jung Kook
- 7 APT.
ROSÉ
- 9 I'll Be There
Jin

3 스테이션

드라이브 하면서 들으면 좋은 노래

1. 재생바

2. 차트

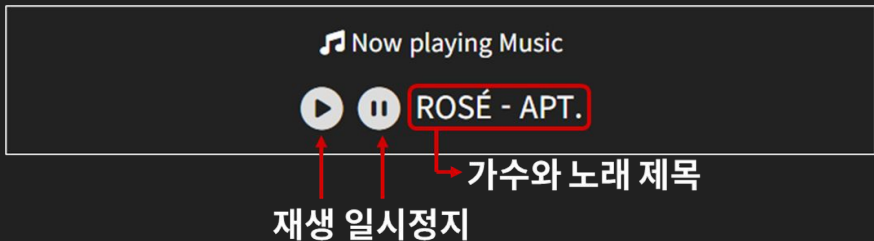
3. 스테이션

• 상황별 맞춤 노래가 저장됨

드라이브 하면서 들으면 좋은 노래

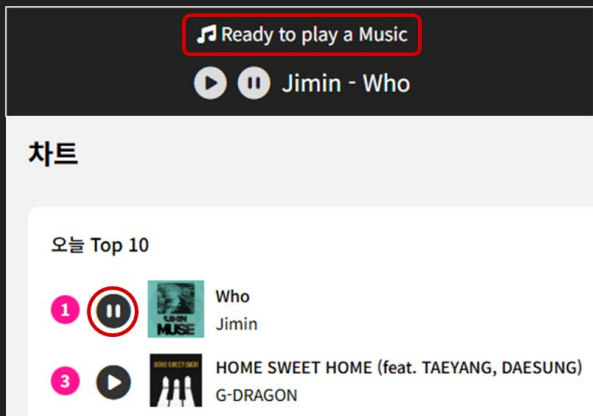
- 1 Picnic
BOL4
- 2 BBIBBI
IU
- 3 Traffic light
Lee Mujin
- 4 Drive
MIYEON
- 5 Travel
BOL4
- 6 sweet little kitty
cherryfilter
- 7 Hold my hand
IU

1. 재생바



- 일시정지 버튼, 재생 버튼
- YouTube IFrame API 사용
- onYouTubeIframeAPIReady 함수 사용하여 노래 재생

• 노래 일시정지



• 노래 재생

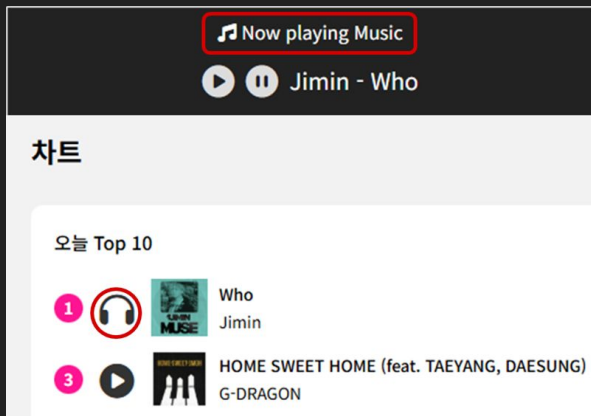


chart.jsp

```
function playOn(){
    checkSessionAndExecute(function() {});
    player.playVideo();
    $(".playing").removeClass("pause");
    $(".state.now").show().siblings(".state.ready").hide();
}

function playPause(){
    checkSessionAndExecute(function() {});
    player.pauseVideo();
    $(".playing").addClass("pause");
    $(".state.now").hide().siblings(".state.ready").show();
}
</script>
```

```
1) SpotifyApi spotifyApi = new SpotifyApi.Builder()
    .setClientId(CLIENT_ID)
    .setClientSecret(CLIENT_SECRET)
    .build();

ClientCredentials credentials = spotifyApi.clientCredentials().build().execute();
accessToken = credentials.getAccessToken();
```

```
2) Playlist playlist = spotifyApi.getPlaylist("", playlistId).build().execute();
```

```
3) for (PlaylistTrack playlistTrack : playlist.getTracks().getItems()) {
    Track track = playlistTrack.getTrack();
    String title = track.getName();
    String artist = track.getArtists()[0].getName();
    String albumImageUrl = track.getAlbum().getImages()[0].getUrl();

    trackInfoList.add(new TrackInfo(title, artist, albumImageUrl));
```

```
4) List<PlaylistTrack> playlistTracks = Arrays.asList(playlist.getTracks().getItems());
    for (PlaylistTrack playlistTrack : playlistTracks) {
        Track track = playlistTrack.getTrack();

        String title = track.getName();
        String artist = track.getArtists()[0].getName();
        String albumImageUrl = track.getAlbum().getImages()[0].getUrl();

        trackInfoList.add(new TrackInfo(title, artist, albumImageUrl));
```

2. 차트

1) Spotify API 접근을 위한 Access Token 생성

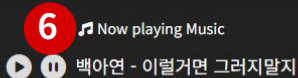
2) Spotify에서 플레이리스트 정보 가져오기

```
https://open.spotify.com/playlist/37i9dQZEVXbNxxXF4SkHj9F
```

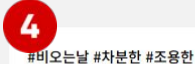
↑ PlaylistID

3) 플레이리스트에서 트랙 정보 추출

4) 트랙 리스트 반환



플레이리스트



1) 해시태그 선택

- 3개까지만 선택
- 3개 초과, 미만 선택할 경우 3개 선택해달라고 경고

2) 뮤직리스트 보기 버튼 클릭

- 선택한 해시태그에 맞는 노래 10곡 랜덤으로 추천

3) 리셋 버튼

- 선택한 해시태그 및 음악 리스트 초기화

4) 추천 노래 10곡 리스트

5) 재생버튼

6) 플레이바

1. 해시태그 리스트

```
<ul class="taglist">
  <li class="tag" data-id="100">#신나는</li>
  <li class="tag" data-id="101">#비오는날</li>
  <li class="tag" data-id="102">#자분한</li>
  <li class="tag" data-id="103">#조용한</li>
  <li class="tag" data-id="104">#파티</li>
  <li class="tag" data-id="105">#운동</li>
  <li class="tag" data-id="106">#명상</li>
  <li class="tag" data-id="107">#휴식</li>
  <li class="tag" data-id="108">#잠들때</li>
  <li class="tag" data-id="109">#노동요</li>
</ul>
```

2. 리스트 출력

```
$.ajax({
  type: 'GET',
  url: '/music/playlist',
  data: {
    'hashtag1': selectedTagArray[0], // 첫 번째 해시태그
    'hashtag2': selectedTagArray[1], // 두 번째 해시태그
    'hashtag3': selectedTagArray[2] // 세 번째 해시태그
  },
  contentType: 'application/json',
  success: function(response) {

    var playlist = response.playlist;

    if (Array.isArray(playlist)) {
      // 선택된 해시태그에 해당하는 데이터만 필터링
      var filteredResponse = playlist.filter(item =>
        selectedTagArray.includes(item.hashtag_name) // selectedTagArray 사용
      );

      var html = '';
      filteredResponse.forEach(function(music) {
        html += '<li class="item">'
        html += '  <button type="button" id="play" class="playBtn" '
        html += '    data-art-name="' + music.art_name + '" '
        html += '    data-music-name="' + music.music_name + '" '
        html += '    onclick="playTrack(this)">';
        html += '  </button>'
        html += '  <div class="arti">'
        html += '    <i class="name">'+music.art_name+'</i>'
        html += '    <i class="tit">'+music.music_name+'</i>'
        html += '  </div>'
        html += '</li>'
      });
    }
  });
});
```

3. Mapper

```
<select id="findALLMusicAjax" parameterType="map" resultMap="musicResultMap">
  SELECT DISTINCT
    m.music_id,
    m.artist_id,
    m.music_name,
    m.track,
    m.playtime,
    a.art_name,
    a.art_img,
    h.hashtag_name,
    h.hashtag_id
  FROM
    music_tbl m
  JOIN
    artist_tbl a ON m.artist_id = a.artist_id
  JOIN
    hashtag_tbl h ON m.music_id = h.music_id
  WHERE
    <if test="hashtag1 != null"> h.hashtag_name = #{hashtag1} </if>
    <if test="hashtag2 != null"> OR h.hashtag_name = #{hashtag2} </if>
    <if test="hashtag3 != null"> OR h.hashtag_name = #{hashtag3} </if>
  GROUP BY
    m.music_id
  LIMIT 10
</select>
```

커뮤니티

당신의 아티스트를 만나보세요!



뉴진스



있지



블랙핑크



에스파



아이브



BTS



1

localhost:8080 내용:

구독중인 아티스트는 블랙핑크 (BLACKPINK)입니다. 블랙핑크 (BLACKPINK) 페이지를 이용해주세요.

확인

2

1. 아티스트 리스트

2. 구독 중인 아티스트 알림

- 구독한 아티스트가 아니면 알림

1. 아티스트 게시물 영역

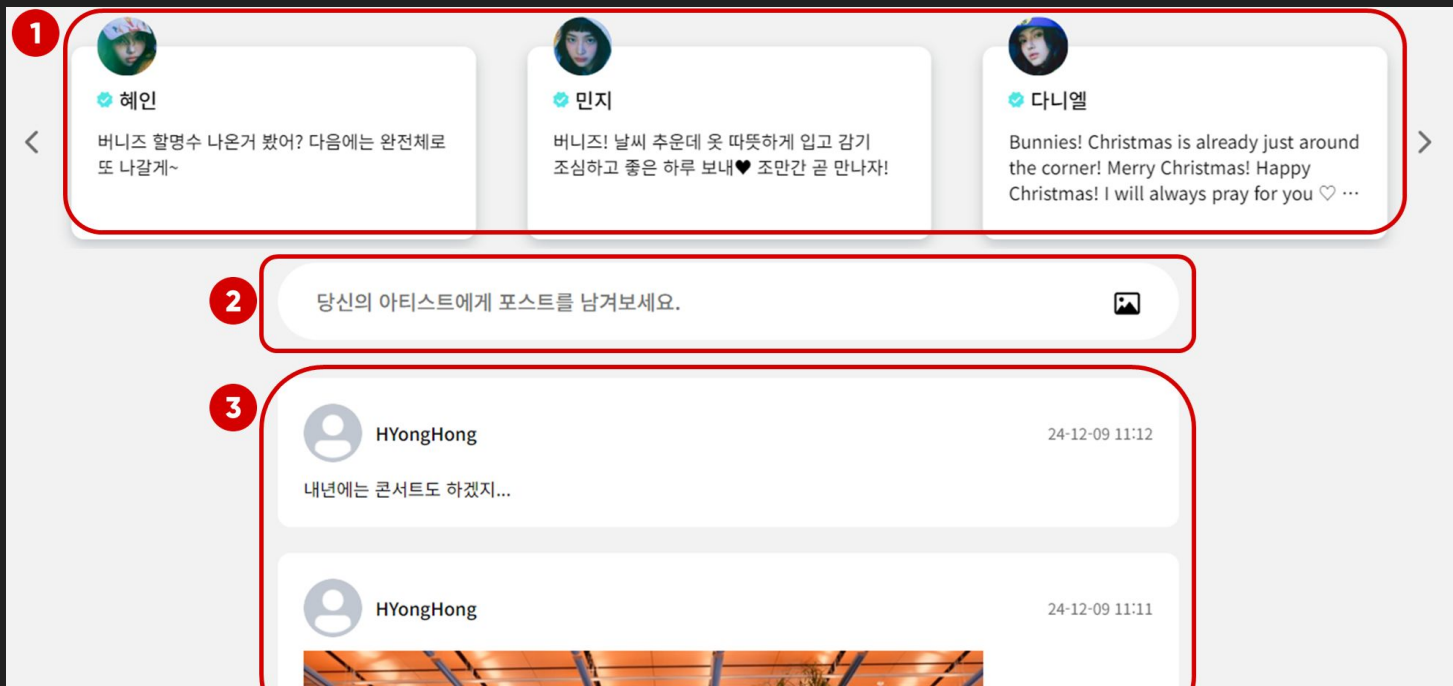
- 카드 형식
- 간편한 탐색

2. 게시물 작성 영역

- 아티스트, 구독자가 서로에게 메시지와 사진 남기기

3. 회원 게시물 영역

- 댓글과 좋아요
- 사진 첨부와 메시지



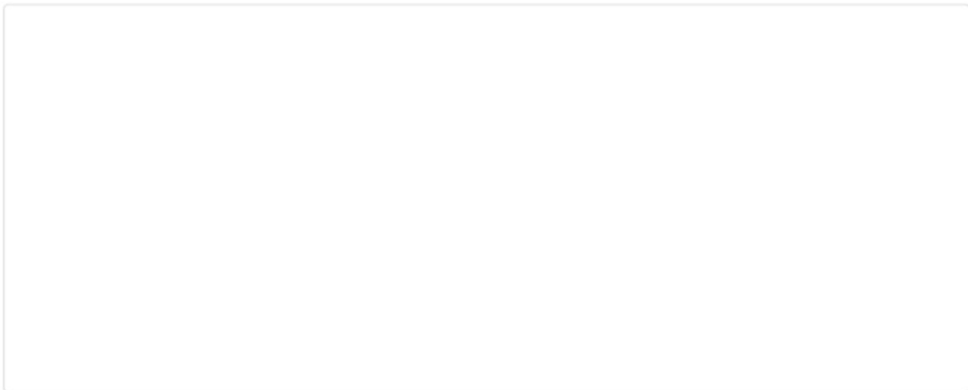
1

당신의 아티스트에게 포스트를 남겨보세요.



2

포스트 쓰기
Newjeans



첨부파일 선택

등록

3



JWonJeong

24-12-09 11:05

뉴진스 덕분에 오늘 하루도 따뜻한 기분으로 보냈어요. 앞으로도 여러분과 함께하는 모든 순간이 기대돼요. 사랑해요!

1. 게시물 작성

- 게시물 작성 UI 클릭 시 게시물 작성 팝업이 나타납니다.

2. 게시물 작성 팝업

- 게시물 작성과 이미지(jpg,png,gif) 첨부 가능

3. 게시물 등록


 헤인 24-12-09 10:52

버니즈 할명수 나온거 봤어? 다음에는 완전체로 또 나갈게~



4개의 댓글
6개의 좋아요

 JieunHong 24-12-09 11:09
다음엔 완전체로!!!!

 SHoJeong 24-12-09 11:03
나만의 개그맨 뉴진스

 HNnaPark 24-12-09 11:00
너무 재밌더라~!!! 애비진스 짱!

 HYongHong 24-12-09 10:58
단체사진 너무 예쁘다

GOOD!!!!!!

좋아요 기능

4개의 댓글
7개의 좋아요



댓글 기능

5개의 댓글
7개의 좋아요



 HYongHong 24-12-10 14:33


GOOD!!!!!!

 JieunHong 24-12-09 11:09

다음엔 완전체로!!!!

 SHoJeong 24-12-09 11:03

나만의 개그맨 뉴진스


 HNnaPark 24-12-09 11:00

너무 재밌더라~!!! 애비진스 짱!

 HYongHong 24-12-09 10:58


단체사진 너무 예쁘다

04 - 5 커뮤니티




 JieunHong

24-12-09 11:09


You got me looking for attention 가끔은 정말 핫갈리지만 분명한 건 Got me looking for attention 어텐션은 진짜 개명곡




2개의 댓글

 HYongHong 24-12-09 11:11  

난 디토


 JieunHong 24-12-09 11:09

난 하입보이가 최애임

댓글을 입력하세요. 

1. 댓글 수정 버튼
2. 댓글 수정창
3. 수정 댓글 반영

2개의 댓글



 HYongHong 24-12-09 11:11  

난 디토 ㅇㅇ


 JieunHong 24-12-09 11:09

난 하입보이가 최애임

2개의 댓글

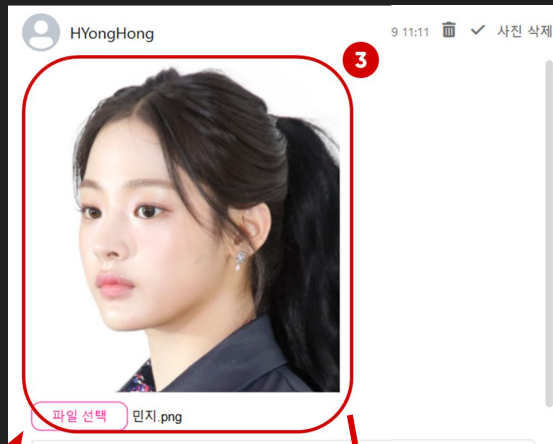
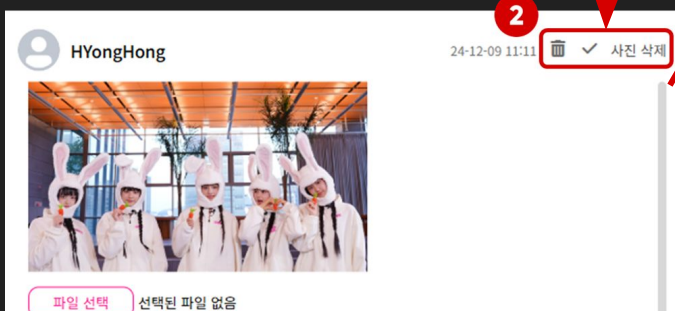
 HYongHong 24-12-09 11:11  

난 디토 ㅇㅇ

 JieunHong 24-12-09 11:09

난 하입보이가 최애임

04 - 5 커뮤니티



1. 게시물 작성, 삭제 버튼
2. 게시물 내용 수정, 첨부 사진 변경
3. 사진 미리보기
4. 게시물 수정

플레이리스트

Chart 페이지로 이동

커뮤니티

알림 ✕

멤버십 level 1 구매 후 이용 가능합니다.

[멤버십 구매하기](#)

알림 ✕

멤버십 level 2 구매 후 이용 가능합니다.

[멤버십 구매하기](#)

멤버십 페이지로 이동

04 - 6 멤버십 결제

Level1 구매

Level 1
단일 스트리밍 이용권
3,900원
구매하기

Level2 구매

Level 2
스트리밍 이용권 + 아티스트 구독
6,900원
구매하기

2. 구독 하고 싶은 아티스트 선택

멤버십 결제

일반결제 pay

1. 멤버십 페이지에서 구매하기 버튼 클릭

멤버십 결제

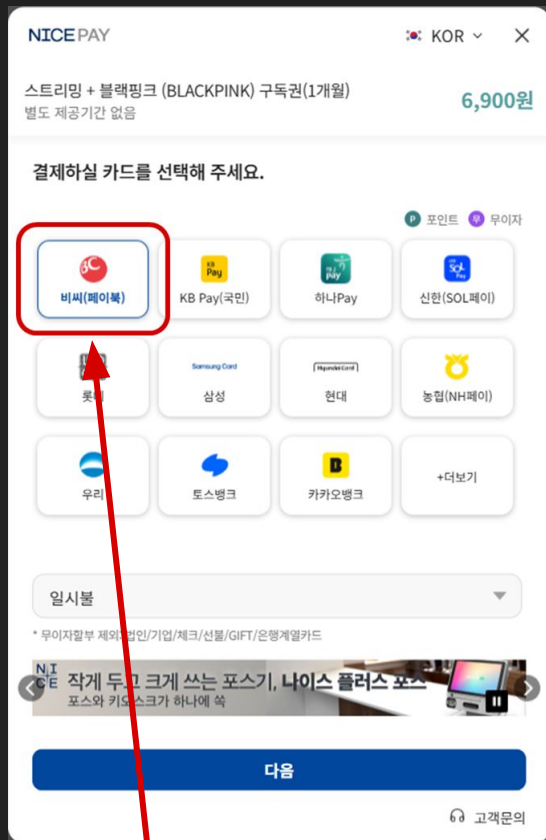
구독 아티스트 선택

BLACKPINK

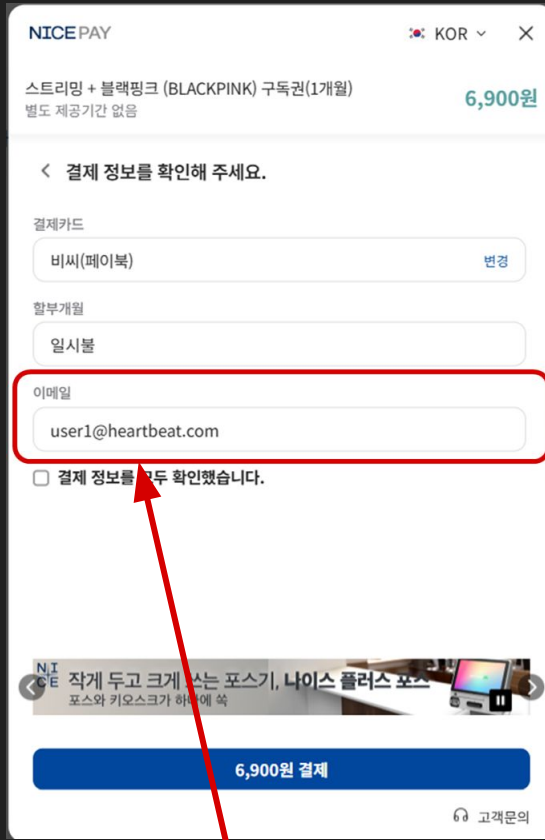
일반결제 pay

- AESPA
- BLACKPINK
- BTS
- IVE
- ITZY
- NewJeans
- NCT-127
- NCT-DREAM
- SEVENTEEN

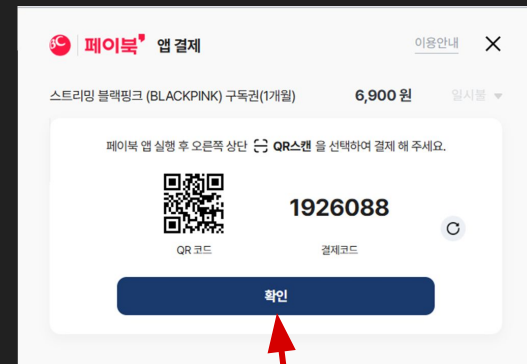
04 - 6 멤버십 결제



3. 결제 방법 선택
(일반결제는 나이스페이먼트 사용)

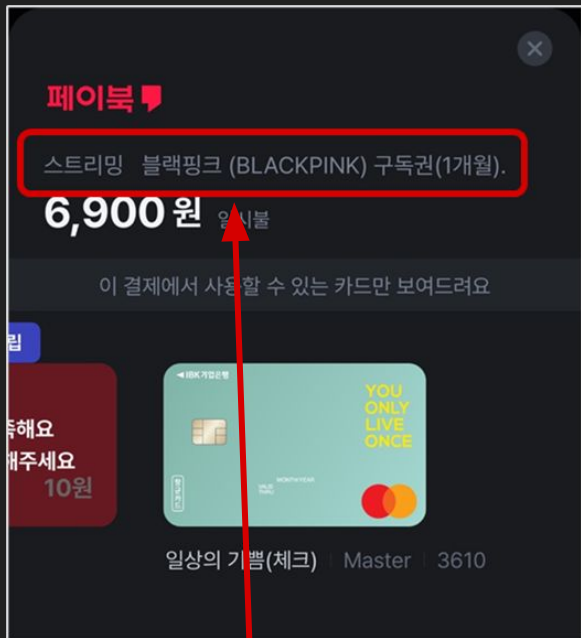


4. 유저 정보를 바탕으로 결제 진행

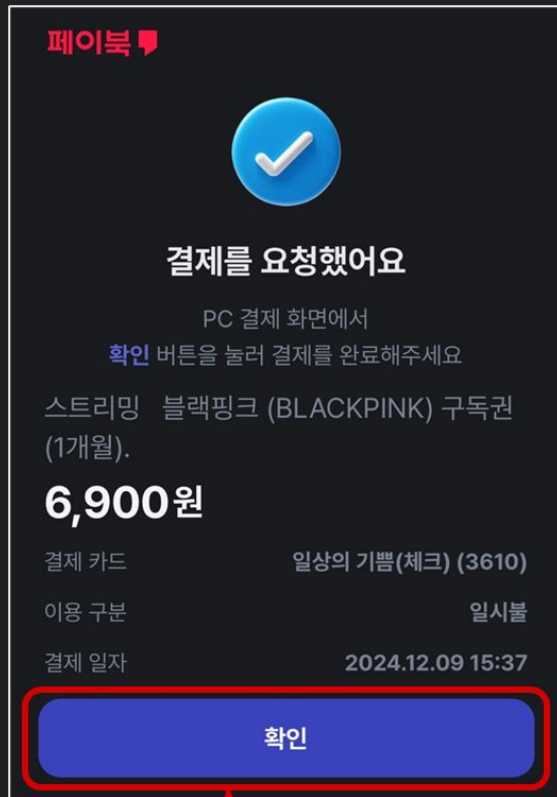


5. 페이북 통해 QR코드 결제

04 - 6 멤버십 결제



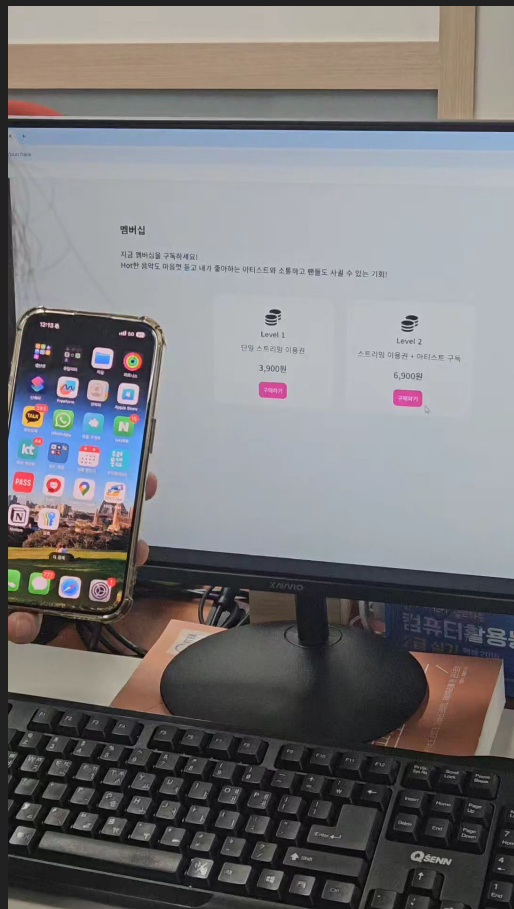
6. 결제하는 상품 확인 후 결제



7. 페이북 앱에서 결제 승인



8. 결제완료

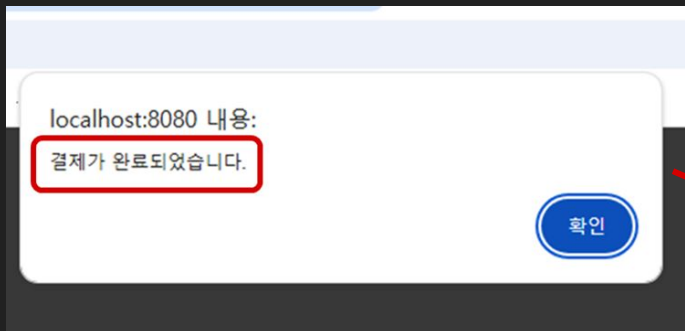


1. 결제 방법 선택 (카카오페이 사용)

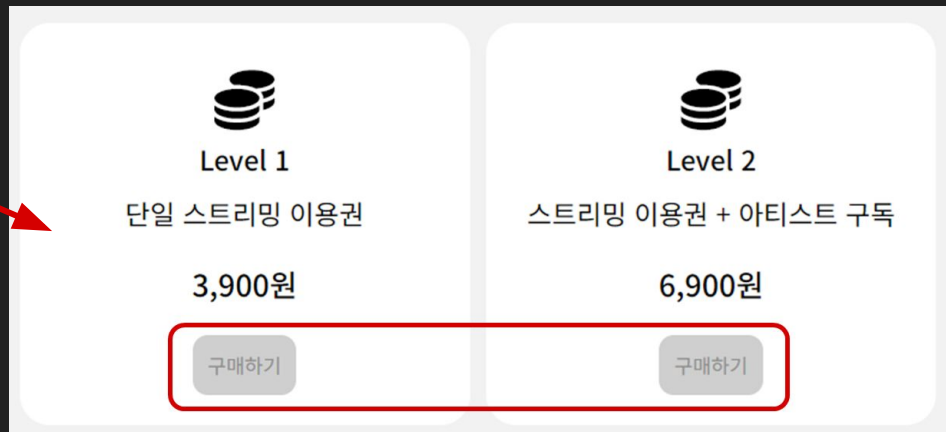
2. 카카오톡 QR코드 사용해서 결제

3. 확인을 누르면 결제 완료

04 - 6 멤버십 결제



6. 결제 완료 창이 뜨면서 바로 멤버십 이용



7. 멤버십 이용자는 결제버튼 이용 불가



8. 마이페이지에서 멤버십 등급 확인 가능

04 - 6 멤버십 결제

포트원을 사용해서 결제 api 구현

결제 api 추가 버튼

수단	결제대행사	채널이름	PG Provider	MID	채널키
결제	카카오페이	카카오페이 일반결제	kakaopay	TCOONETIME	channel-key-7f6d41c7-9201-45c1-b941-2167b39271e4
결제	나이스페이먼츠	나이스페이먼츠 결제창 일반결제-가상계좌	nice_v2	iimport03m	channel-key-bb343318-fb81-4bd2-b9b1-aa96b8a14a3a

포트원 페이지 결제내역에서 결제정보 확인 및 취소 가능

결제상태	UID/MID	주문명	고객정보	결제수단	결제대행사-라우팅그룹	결제금액	구분	상태 승인 시각
결제완료	imp_233981123792 nobody_1733722981022	스트리밍 + 블랙핑크 (BLACKPINK) 구독권(1개월)	김홍현	체크카드	(신) 나이스페이먼츠	6,900 KRW	테스트 일반	2024.12.09 14:43:59

결제 상태 확인

주문 아이템 확인

결제수단 확인

결제 대행사 확인

결제금액 확인

결제 취소 버튼

04 - 7 마이페이지 - 내 정보 변경

중복확인 X

abc은/는 사용 가능합니다.

확인

회원 탈퇴

이메일 입력

비밀번호 입력

비밀번호 확인

탈퇴하기 달기

마이페이지

내 정보 변경 멤버십 변경 내 작성글 확인

1 비밀번호 변경

현재 비밀번호 입력

새 비밀번호 입력

새 비밀번호 확인

2 닉네임 변경

닉네임 입력 중복확인

3 프로필 사진 변경

사진 변경 선택된 파일 없음

프로필 사진 초기화

4 회원 탈퇴

탈퇴하기

1. 비밀번호 변경

- 1) 새 비밀번호 입력값 서로 일치 확인
- 2) 현재 비밀번호 DB와 비교
- 3) 새 비밀번호 DB에 저장

2. 닉네임 변경

3. 프로필 사진 변경

4. 회원 탈퇴

- 1) 사용자 입력값과 DB 비교
- 2) 사용자의 정보 탈퇴유저테이블로 옮기기
- 3) 사용자의 정보 유저테이블에서 삭제

04 - 7 마이페이지 - 내 작성글 확인

마이페이지

[내 정보 변경](#) [멤버십 변경](#) [내 작성글 확인](#)

401 여러분 벌써 크리스마스네요 다들 오늘 하루 행복하게 보내세요~~

1

내 작성글 확인을 통해 커뮤니티에서 내가 작성한 글을 확인하고 관리가 가능

04 - 7 공지 및 문의

공지 및 문의 ¹

전체 공지 및 문의 **내 문의글 확인** ²

제목

[공지사항] 고객센터 운영 시간 변경 안내	관리자
[공지사항] 정기 점검 안내	관리자
[공지사항] 신규 서비스 오픈 안내	관리자
[공지사항] 업데이트 예정 안내	관리자
[공지사항] 신규 기능 업데이트	관리자
게시물 사라짐	SHuiPark
구독 결제 문의	JMinHan
커뮤니티 접속	HyeonSooKim
아티스트 구독 변경	JWonJeong

1

1. 공지 및 문의 페이지
2. 내가 작성한 문의글 확인
3. 게시물 상세페이지로 이동

04 - 7 공지 및 문의

관리자

[공지사항] 고객센터 운영 시간 변경 안내

2024-12-09 10:49

[공지사항] 고객센터 운영 시간 변경 안내 안녕하세요, 회원님! 항상 저희 HeartBeat를 사랑해주셔서 감사합니다. 회원님들의 불편을 최소화하고자 다음과 같이 고객센터 운영 시간이 변경됩니다. 변경 전: 평일 10:00~17:00 변경 후: 평일 09:00~18:00 감사합니다.

4

수정하기

삭제

목록

5

답변은 바로 주시나요?

등록

전체 공지 및 문의 **내 문의글 확인**

제목

검색어 입력

검색

6

전체선택

삭제

62 [공지사항] 고객센터 운영 시간 변경 안내

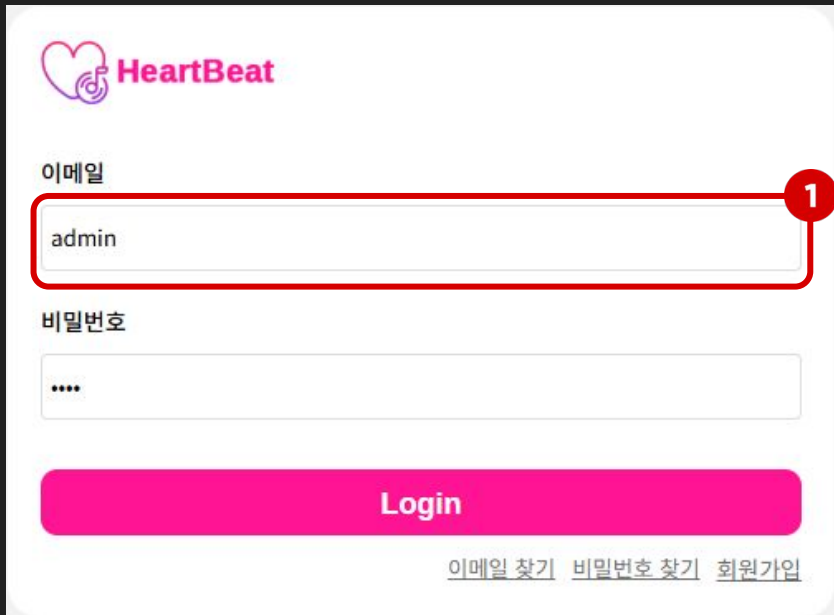
61 [공지사항] 정기 점검 안내

60 [공지사항] 신규 서비스 오픈 안내

46 [공지사항] 업데이트 예정 안내

45 [공지사항] 신규 기능 업데이트

04 - 8 관리자 페이지



HeartBeat

이메일

admin

비밀번호

....

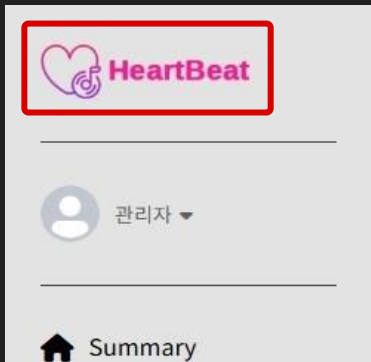
Login

[이메일 찾기](#) [비밀번호 찾기](#) [회원가입](#)

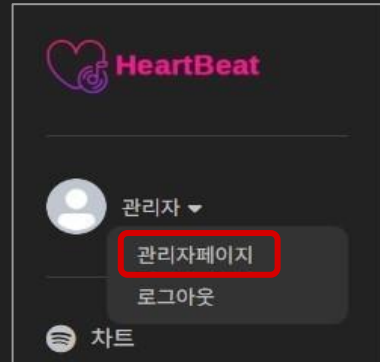
1. 어드민 페이지

- 아이디 admin으로 로그인 시 관리자 페이지로 연결

[관리자페이지]



[유저 페이지]



2. 페이지 간 이동

04 - 8 관리자 페이지 - Summary

요약정보

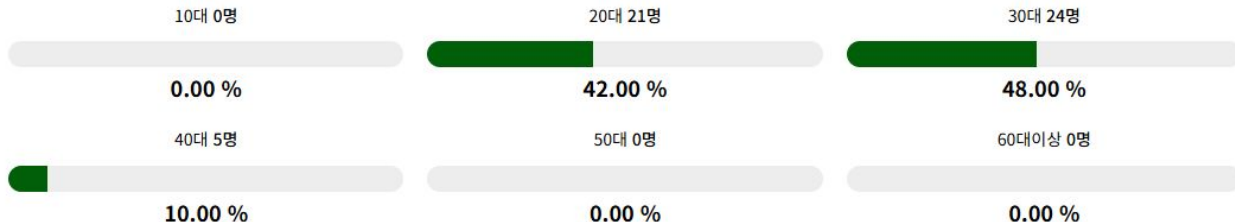
- 오늘 가입한 유저 : 0
- 오늘 탈퇴한 유저 : 0
- 회원 총 인원: 50
- 가장 많은 구독자를 보유한 아티스트 : 뉴진스 (NewJeans) 구독자 : 7

• 회원 등급 분석

총 구독자 수 : 46



• 회원 연령대별 분석



요약정보

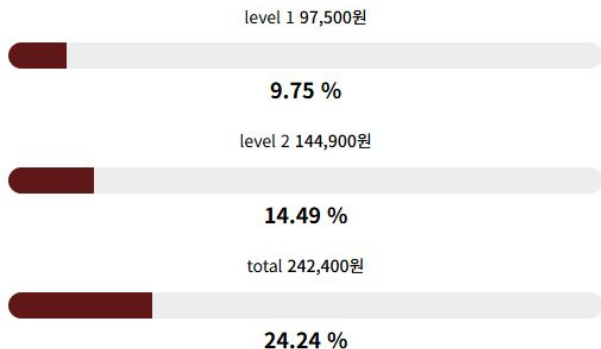
- 주요 데이터를 실시간으로 시각화

- 1) 오늘 가입한 유저
- 2) 오늘 탈퇴한 유저
- 3) 총 회원 수
- 4) Best Artist
- 5) 회원 등급 분석
- 6) 회원 연령대별 분석

04 - 8 관리자 페이지 - Summary

• 매출 분석

목표 금액 :1,000,000원



• 좋아요가 가장 많은 게시물 top5

• 게시물 번호 401 | 작성자 예지 | 좋아요 11

• 게시물 번호 400 | 작성자 유나 | 좋아요 10

• 게시물 번호 404 | 작성자 류진 | 좋아요 10

• 게시물 번호 402 | 작성자 류진 | 좋아요 9

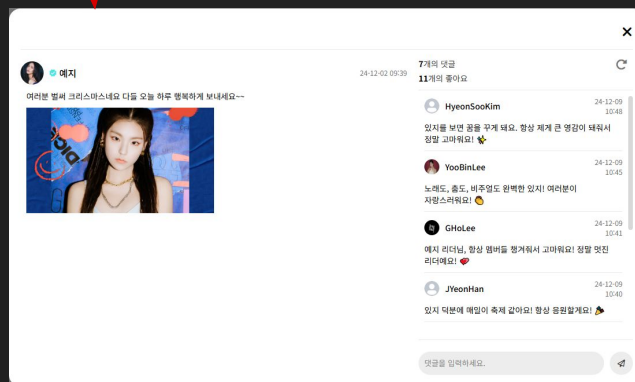
• 게시물 번호 403 | 작성자 리아 | 좋아요 9

요약정보

- 주요 데이터를 실시간으로 시각화

7) 매출분석

8) 가장 인기 많은 게시물



04 - 8 관리자 페이지 - 회원 리스트

* 필수 입력 사항

* 이메일 중복확인

* 비밀번호

* 비밀번호 확인

* 이름

뉴진스 (NewJeans) ▾

생년월일 연도-월-일

* 휴대폰번호 중복확인

* 닉네임 중복확인

등급 해당 없음 ▾

구독기간 연도-월-일 ~ 연도-월-일

직업 ▾

계정생성

localhost:8080 내용:
이메일을 입력하세요.

확인

- ## 회원리스트 > 신규등록
- 1) 관리자가 계정 직접 생성
 - 2) 유효성 검증 완화
 - 3) 입력폼 안내 메시지

구독		Subscription_tbl		
no	구독번호	int		NOT NULL
구독_등급	level2	TINYINT		NOT NULL
이메일	email	VARCHAR(50)		NOT NULL
아티스트_아이디	artist_id	INT		NOT NULL
구독_시작날짜	start_date	DATE		NOT NULL
구독_종료날짜	end_date	DATE		NOT NULL

유저		user_tbl		
이메일	email	VARCHAR(50)		NOT NULL
아티스트_아이디	artist_id	INT		NOT NULL
비밀번호	pwd	VARCHAR(20)		NOT NULL
이름	name	VARCHAR(40)		NOT NULL
닉네임	nickname	VARCHAR(20) / UK		NOT NULL
생년월일	birth	DATE		NOT NULL
휴대전화	phone	VARCHAR(13)		NOT NULL
프로필사진	profile_img	VARCHAR(100)		NULL
구독_등급	level	INT		NULL
생성일	reg_date	DATETIME		NOT NULL
수정일	up_date	DATETIME		NOT NULL

유저_역할(ur_)		user_role_tbl		
유저역할	ur_key	INT(AUTO_INCREMENT)		not null
이메일	email	VARCHAR(50)		NOT NULL
역할_아이디	role_id	INT		NOT NULL

흐름도

입력 폼 → Controller → Service → persistence → Mapper → DB 삽입 → 결과 페이지

```
//계정생성
@PostMapping("/adminjoin")
public String adminInsertUser(UserVO userVO,
    @RequestParam("role_id") int role_id,
    @RequestParam(value = "start_date", required = false) String start_date,
    @RequestParam(value = "end_date", required = false) String end_date,
    RedirectAttributes redirectAttributes) {
    System.out.println("==== Controller member(admin) email: " + userVO.getEmail());

    //비밀번호 암호화
    String pwd = userVO.getPwd();
    String encodePwd = bCryptPasswordEncoder.encode(pwd);
    System.out.println("==== Presentation member pwd : " + encodePwd);
    userVO.setPwd(encodePwd);

    // SubscriptionVO 생성 (구독 정보가 있을 경우에만)
    SubscriptionVO subscriptionVO = null;
    if (userVO.getLevel() > 0 && start_date != null && end_date != null) {
        subscriptionVO = new SubscriptionVO();
        subscriptionVO.setEmail(userVO.getEmail());
        subscriptionVO.setLevel(userVO.getLevel());
        subscriptionVO.setArtist_id(userVO.getArtist_id()); // 기본값 0
        subscriptionVO.setStart_date(start_date);
        subscriptionVO.setEnd_date(end_date);
    }

    // 서비스 호출
    int result = service.insertUser(userVO, role_id, subscriptionVO);

    // 결과에 따른 리다이렉트 처리
    if (result > 0) {
        return "redirect:/admin/member"; // 회원가입 성공 시 이동
    } else {
        redirectAttributes.addFlashAttribute("errorMessage", "계정 생성에 실패했습니다.");
        return "redirect:/admin/adminjoin"; // 회원가입 실패 시 이동
    }
}
```

[Controller]

```
//계정생성
@Override
@Transactional // 트랜잭션 처리
public int insertUser(UserVO userVO, int role_id, SubscriptionVO subscriptionVO) {
    // 1. user_tbl에 데이터 삽입
    int result = persistence.insertUser(userVO);

    if (result > 0) {
        // 2. user_role_tbl에 데이터 삽입
        UserroleVO userRole = new UserroleVO();
        userRole.setEmail(userVO.getEmail());
        userRole.setRole_id(role_id);
        persistence.insertUserRole(userRole);

        // 3. subscription_tbl에 데이터 삽입 (구독 정보가 있을 경우)
        if (subscriptionVO != null) {
            persistence.insertSubscription(subscriptionVO);
        }
    }

    return result;
}
```

[Service]

회원 리스트

회원 리스트 직원 리스트

이름



검색어 입력

아티스트 일반 유저

검색

이름 : 리사

닉네임 : 리사

핸드폰번호 : 1111

가입일 : 2024-11-28 16:57:51.0

이메일 : lisa

생년월일 : 1997-03-27

구독등급 : 2

신규등록

수정

삭제

1. 회원 리스트, 직원 리스트를 분리하여 관리
2. 검색 필터를 통해 원하는 정보를 쉽고 빠르게 검색
3. 신규 등록, 수정, 삭제를 통해 회원 관리

04 - 8 관리자 페이지 - 회원 리스트

lisa 님 회원정보 수정

이름
리사

email
lisa

닉네임
리사

생년월일
1997-03-27

연락처
1111

구독등급
2

아티스트 아이디
20119

최근 수정일
2024-12-06 09:28:28.0

회원수정 닫기

회원 수정

- 1. email은 readonly 태그를 사용하여 수정 불가 처리
- 2. 최근 수정일은 disabled 태그를 사용하여 직접 수정 불가 처리

04 - 8 관리자 페이지 - 회원 리스트

게시글 확인

게시물 번호 ▼

검색어 입력

아티스트

일반 유저

검색

1

게시물 번호 : 569

작성자 : JSuhPark

작성일 : 2024-12-09 10:56:00

내용 :

제니아, 네가 우리에게 보여주는 사랑과 관심에 너무 감사해! ❤️ 항상 너의 마음을 느낄 수 있어서 행복해. 앞으로도 계속 함께 걸어가지!

2

보기

삭제

1. 검색 필터
 2. 게시글 보기, 삭제
- 유저 페이지와 동일한 팝업



04 - 8 관리자 페이지 - 회원 리스트

공지 및 문의

전체 공지 및 문의 **내 문의글 확인**

제목 검색어 입력

[공지사항] 고객센터 운영 시간 변경 안내	관리자
[공지사항] 정기 점검 안내	관리자
[공지사항] 신규 서비스 오픈 안내	관리자
[공지사항] 업데이트 예정 안내	관리자
[공지사항] 신규 기능 업데이트	관리자
게시물 사라짐	SHuiPark
구독 결제 문의	JMinHan
커뮤니티 접속	HyeonSooKim
아티스트 구독 변경	JWonJeong

1

2

1. 내 문의글만 볼 수 있게 분리
2. 관리자가 등록한 문의글과 일반 게시글 차별화
3. 일반 게시글 클릭 시 댓글 작성 가능

3

JMinHan
구독 결제 문의
2024-12-09 11:30

멤버십을 결제하고 싶은데 카카오페이 다른 결제 방식은 없나요?

관리자
24-12-09 11:39

안녕하세요, 회원님! 항상 저희 HeartBeat를 사랑해 주셔서 감사합니다. 현재 준비되어있는 결제방식은 카카오페이 유일하며 추후 업데이트 될 예정입니다. 감사합니다. 앞으로 더 노력하는 HeartBeat가 되겠습니다.

수정 삭제

댓글 내용을 입력하세요

05 주요 기능 시연

06 수정 및 발전 방향

UPDATE

1. 관리자 공지사항 댓글 삭제
2. 관리자 팝업 구분
3. 사용자 신고 및 알림 기능

NEW

1. 관리자 요약정보 구글 차트 API 도입
2. CKEditor 도입
3. 메시지 기능

감사합니다.