



## Jelly Music

‘알고리즘의 선택’이라는 말이 생길 만큼 이제는 알고리즘이 단순한 추천을 넘어서 숨어있던 콘텐츠를 유명하게 만드는 역할까지 하고 있습니다. 음악 추천 사이트에서 시작한 아이디어는 음악 플레이 뿐만 아니라 음원, 앨범, 아티스트 정보 제공과 사용자들끼리의 소통을 위한 댓글도 추가하였습니다.

Git Link is here

go!

문서 드라이브 링크 바로가기

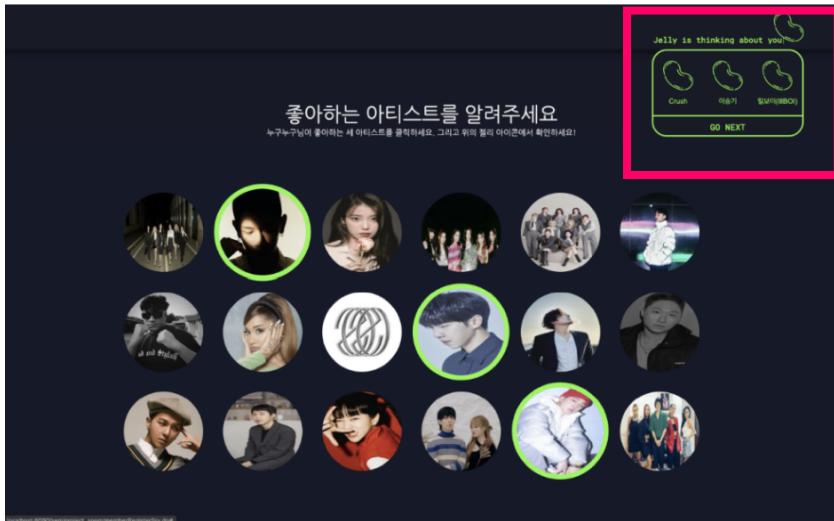
# 취향 선택 페이지



체크박스 카운트를 통해 반드시 세 개를 체크 해야 다음 화면으로 진행할 수 있는 버튼이 활성화 되게 하였습니다.

```
var maxChecked = 3;
var totalChecked = 0;
function CountChecked(field) {
  var value_list = []
  $('input:checkbox[name=achk]:checked').each(function() {
    var checked_value = $(this).val();
    value_list.push(checked_value);
  });
  if (field.checked) {
    totalChecked += 1;
  } else {
    totalChecked -= 1;
  }
  console.log(totalChecked);
  if (totalChecked > maxChecked) {
    alert("최대 3개까지 가능");
    field.checked = false;
    totalChecked -= 1;
  }
}
if (totalChecked == 0) {
} else if (totalChecked == 1) {
  $("#mu_genre_1").val(value_list[0]);
} else if (totalChecked == 2) {
  $("#mu_genre_1").val(value_list[0]);
  $("#mu_genre_2").val(value_list[1]);
} else if (totalChecked == 3) {
  $("#mu_genre_1").val(value_list[0]);
  $("#mu_genre_2").val(value_list[1]);
  $("#mu_genre_3").val(value_list[2]);
}
```

체크 시 어떤 항목이 체크 되었는지 상단 오른쪽에서 확인 할 수 있게 value를 보여주었고, 해당 value 들을 DB에 insert 해주었습니다.



# 메인 페이지



**Jelly Music**

- Hello, aaab
- 마이페이지
- 노래
- 마이플레이리스트
- 나만의 추천
- 고객지원센터
- 이용권구매

Player

aaab님을위한추천곡

Blueming  
아이유(IU)

Blue & Grey  
방탄소년단

Stay  
방탄소년단

VVS (Feat. JUSTHIS) (Prod. GroovyRoom)  
미란이

도망가  
MINO(승민호)

더보기

aaab님이좋아할수도있는아티스트

방탄소년단

Ariana Grande(아리아나 그란데)

aespa

방탄소년단

방탄소년단

미란이

aaab님이최근재생한음악

Love, Open

TAKE  
MINO

Love, Open

Love, Open

SHOW ME THE MONEY 9  
Episode 1

더보기



## 회원이 최근 들은 음악 리스트

```
// 회원이 최근 들은 음악 리스트
public List<MainVO> getMemPlay(Connection conn, String mem_id) {
    List<MainVO> mvo = new ArrayList<MainVO>();
    String sql = "select * from (select rownum rnum, m.* "
        + "from (select mem_id, mu_name, al_name, art_name, play_date, f_name, music.mu_no "
        + "from music join album on music.al_no = album.al_no "
        + "inner join artist on music.art_no = artist.art_no "
        + "left outer join allfile on album.f_no = allfile.f_no "
        + "right outer join musichistory on music.mu_no = musichistory.mu_no "
        + "where mem_id=? and play_now='Y' order by play_date desc) m) "
        + "where rnum between 1 and 5";
```

회원이 최근 들은 음악 리스트 쿼리문입니다.  
rownum을 사용하여 1부터 5까지의 데이터를 조회하였고,  
음원 이름을 위한 music 테이블과 앨범 이름을 위한 album 테이블, 아티스트 이름을 위한 artist  
테이블, 앨범 사진을 위한 allfile 테이블을 조인 하였으며,  
플레이리스트에 추가 유무를 판단하기 위한 musichistory 테이블을 조인하였습니다.



## 회원별 추천 아티스트 리스트

```
// 회원별 추천 아티스트
public List<MainVO> mayLikeArt(Connection conn, String mem_id) {
    List<MainVO> mvo = new ArrayList<MainVO>();
    String sql = "select * from "
        + "(select * from (select rownum rnum, m.* "
        + "from(select art_name, f_name, mu_genre "
        + "from artist join allfile on artist.f_no = allfile.f_no "
        + "inner join music on artist.art_no = music.art_no "
        + "group by mu_genre, art_name, f_name) m) ) a "
        + "where mu_genre in (select distinct b.mu_genre from "
        + "(select mu_genre, count(mu_genre), art_name, f_name, mem_id "
        + "from artist inner join music on artist.art_no = music.art_no "
        + "inner join allfile on artist.f_no = allfile.f_no "
        + "inner join musichistory on musichistory.mu_no = music.mu_no "
        + "group by mu_genre, f_name, art_name, mem_id "
        + "having mem_id = ? "
        + "order by count(mu_genre) desc) b)";
```

*회원별 추천 아티스트 리스트 쿼리문입니다*

*회원이 들은 음악의 장르를 조회한 뒤 해당 장르와 같은 음악을 많이 낸 아티스트를 count로 조회하였고, 그 수가 큰 아티스트부터 추천하였습니다.*

*해당 쿼리문에는 오류 사항이 많아 추후 수정할 예정입니다.*



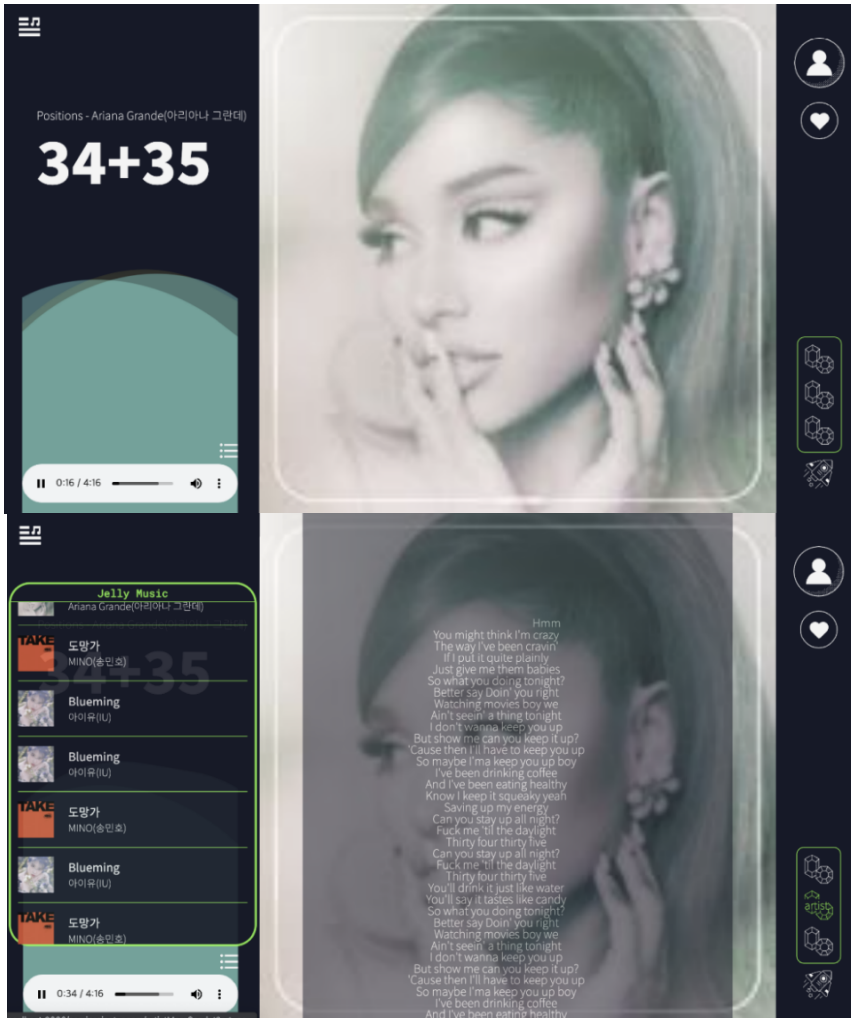
## 회원별 추천 음악 리스트

```
ing sql = "select * from"
+ "(select rownum rnum, s.* from "
+ "(select (nvl(m.cnt, 0) + nvl(h.cnt, 0)) sum, nvl(h.mu_no, m.mu_no) mu_no1 from "
+ "(select count(mem_id) cnt, mu_no from musichistory "
+ "where mem_id in(select mem_id from musichistory "
+ "where mu_no =(select mu_no from (select rownum rnum, m.* from "
+ "(select * from musichistory where mem_id = ? order by play_more_min desc) m) "
+ "where rnum=1)) group by mu_no) m full outer join (select count(mem_id) cnt, mu_no from heart
+ "where mu_no in(select mu_no from heart where mem_id = ?) group by mu_no) h "
+ "on m.mu_no=h.mu_no order by sum desc) s where mu_no1 not in (select mu_no from heart "
+ "where mem_id = ?) order by sum desc)where rnum<6";
```

*회원별 추천 음악리스트 쿼리문입니다*

*회원이 많이 들은 음악과 하트를 누른 음악을 조회하여 추천하였습니다.*

*해당 쿼리문에는 오류 사항이 많아 추후 수정할 예정입니다.*



## 이용권 유무 검사

```
var vid = document.getElementById("player");
vid.onplaying = function() {
  timeout = setTimeout(function() {
    vid.pause();
    alert("이용권을 구매해주세요");
    window.opener.location.href = "<%=request.getContextPath() %>/MembershipServlet"
  }, 3000);
};
```

플레이어 진입 시 이용권 유무를 검사하여 1분 미리 듣기를 구현하였습니다.  
시현 당시 시간상 3초로 설정하여 시현하였습니다.  
이용권이 없을 시 setTimeout을 사용하여 음악을 멈추고,  
멤버십 구매 페이지로 이동하게 구현하였습니다.