

# PORTFOLIO

성실함과 책임감을 최우선으로 생각하겠습니다.  
개발자 백성곤입니다.



## ■ 프로필

Name 백성곤

MOBILE 010-8814-0575

E-MAIL canoa0327@naver.com

ADDRESS 경기 하남시 망월동

## ■ 보유 기술

- C - React
- Java - JSP
- Python - Oracle
- html/css - Javascript
- vue.js

Baek SeongKon  
95.03.27

## ■ 학력

2011.3 – 2014.2 하남정보산업고등학교 졸업

2014.3 – 2020.2 한신대학교 컴퓨터공학부 졸업

## ■ 프로젝트

Team Project [JSP] 예술의전당 홈페이지 구현

Single Project [Vue.js] VueTube

## 1ST PROJECT

# 예술의전당 웹사이트 구현

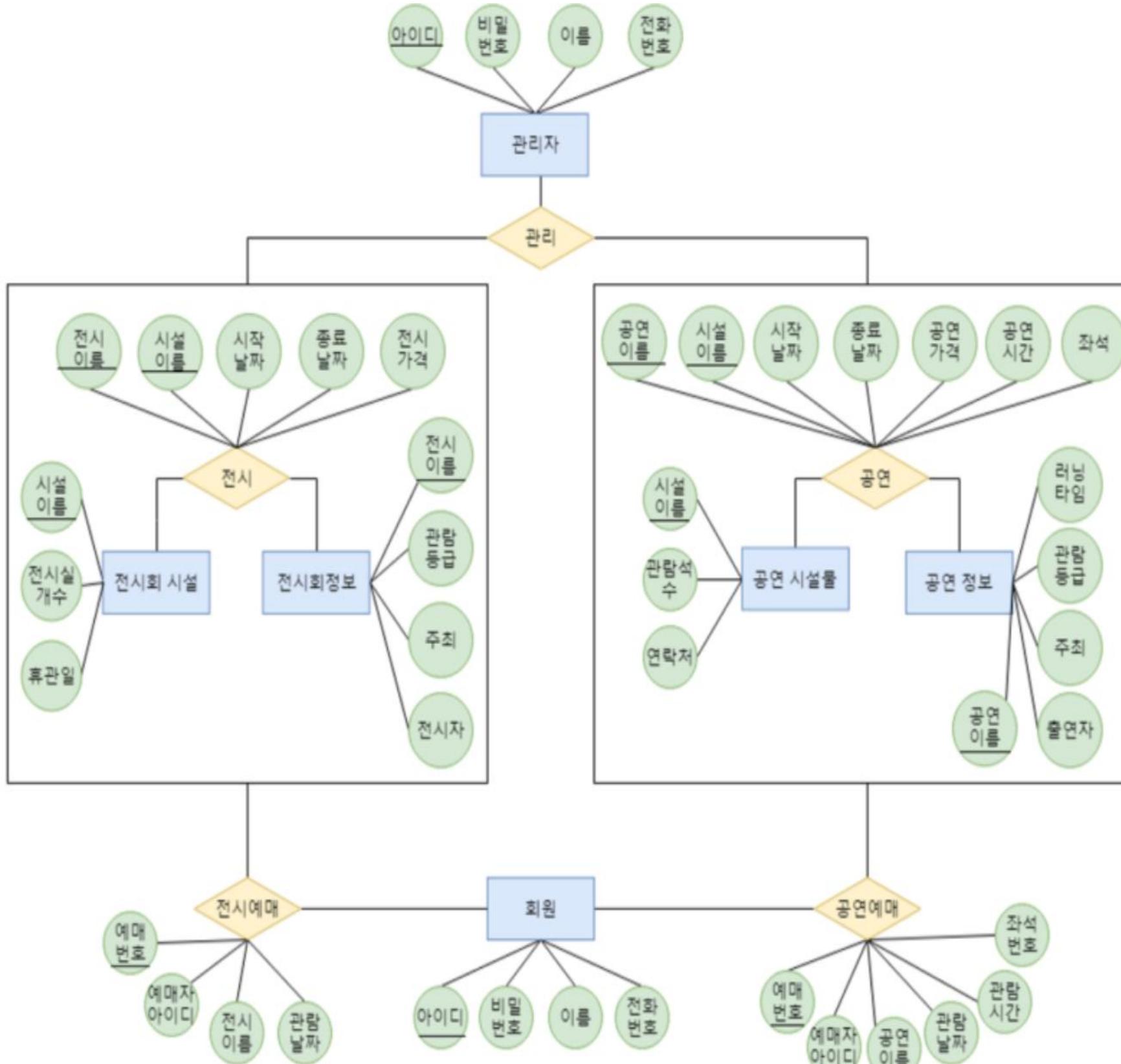
### 소개 및 기능

- 사용자는 웹사이트를 이용하여 전시회 일정, 전시장 등의 정보 조회 가능.
- 원하는 전시회나 공연 등의 예매 기능과 취소 기능 제공
- 관리자는 웹사이트를 이용하여 전시회 일정, 전시장 등의 정보 수정 가능.

### 개발 환경 및 사용 기술

- Eclipse
- Java
- JSP
- html/css
- MySQL
- Tomcat

# ER 다이어그램 / 테이블 구성



Tables_in_artmoa	
+	admin
+	exh_hall
+	exh_info
+	exh_reserve
+	exhibit
+	member
+	per_hall
+	per_info
+	per_reserve
+	perform

- 데이터베이스 테이블

# DB 구성(admin, member)

adminid	adminpw	adminname	admintel
AD100	1111	나수영	01011111111
AD200	2222	이찬수	01022222222
AD300	3333	정기태	01033333333

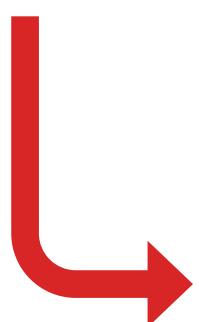
- admin 테이블

- 해당 DB의 값으로 로그인을 요청하면,  
관리자 권한으로 제어할 수 있다.

memid	mempw	memname	memtel
ME100	804073CEA840D2433DB3E494AA3DEC6C	백성곤	01011111111
ME200	EEF523A9E9BFB40077241D34B8B80C94	사원기	01022222222
ME300	25499B1F02F15D04EBF63B10BA96BCF4	신혜림	01033333333
ME400	A12321A124680C22B81A578713B8D0DD	정민지	01044444444
ME500	9DD48C56838F4960C94E6290006B15EA	송병길	01055555555
ME600	DC2F71C9F1EDE26D09118FFD3D7E7DB2	박종화	01066666666

- member 테이블

- 패스워드는 쉽게 알 수 없도록  
AES 암호화 기법 적용



```
HEX(AES_ENCRYPT('1111', 'key'))  
HEX(AES_ENCRYPT('2222', 'key'))  
HEX(AES_ENCRYPT('3333', 'key'))  
HEX(AES_ENCRYPT('4444', 'key'))  
HEX(AES_ENCRYPT('5555', 'key'))  
HEX(AES_ENCRYPT('6666', 'key'))
```

## DB 구성(exh, per)

[ exh\_hall  
exh\_info  
exh\_reserve  
exhibit  
per\_hall  
per\_info  
per\_reserve  
perform

- exh 테이블 : 전시회 DB
  - per 테이블 : 공연 DB
- \_hall : 전시장, 공연장의 DB  
\_info : 전시회, 공연의 DB  
\_reserve : 전시회, 공연을 예약한 회원의 DB
- exhibit : 전시회 일정에 대한 DB  
perform : 공연 일정에 대한 DB

# 메인페이지

## ArtMOA

MAIN 공연 전시 공연장 전시장 예매 관리자로그인

### What's On

**한화 클래식 2018**  
안드레아스 슬과 영국의 잉글리시 콘서트를 초대했다.  
영국과 이탈리아, 두 개의 주제에 맞춘 프로그램으로 구성되어 있다.  
윌리엄 보이스와 찰스 에이비슨의 음악도 함께 들어 볼 수 있다.  
그 밖에도 비발디, 토렐리와 같은 이탈리아 작곡가들의 작품도 연주한다.

[더보기](#)

**제8회 대한민국발레축제 BALLET FESTIVAL KOREA**  
국립발레단, 유니버설발레단, 서울발레시어터, 김지안 발레단  
정형일 Ballet Creative, 임혜경 Le Ballet, 윤전일 Dance Emotion  
기획공연 | 초청안무가 김용걸, 김세연

[더보기](#)

**예술의전당 어린이 가족 페스티벌(7/20~9/2)**  
똘방이와 리나 (07.20-07.29)  
평강공주와 온달바보 (07.31-08.12)  
빅토리아의 100번째 생일 (08.14-08.26)  
피노키오 (08.23-09.02)

[더보기](#)

### 예술의전당 오시는 길

원활한 관람을 위해  
가급적 대중교통을 이용 바랍니다.

	지하철	
3호선	남부터미널역 5번 출구	
2호선	서초역 3번 출구	
4호선	사당역 1번 출구	

	버스	
간선	406, 405	
지선	5413	
직행	1550-2, 1553	
마을	서초11, 서초17, 서초22	

	승용차	
경부고속도로 서초 IC 예술의전당 방향 → 남부순환로 사당방면으로 직진 → 예술의전당앞 교차로 좌측에 예술의 전당		

# 공연 & 전시 페이지 데이터바인딩

The top screenshot shows a page for '리오 삼바달 & 경기필하모닉' (Rio Sambad & Kyunggi Philharmonic) featuring a portrait of a man and a large teal 'G'. The bottom screenshot shows a page for the '2018 예술의전당 개관 30주년' (2018 Seoul Arts Center 30th Anniversary) with a large yellow '30' logo and a pink image of a person's legs.

**Event Page Data:**

전시명	장소	시작날짜	끝날짜	가격
명재운중	서울시예박물관	20180329	20180513	15000원
서울모던아트쇼	한가람미술관	20180504	20180508	10000원
에이엔씨아트페스티벌2018	그린나래미술관	20180424	20180429	10000원
아시원 개인전, 아름다운 거울	그린나래미술관	20180506	20180514	13000원
캘리는 나의 삶	한가람미술관	20180414	20180421	0원
향부용 헬링 그레이피즘 - 춤추는 사람들	나르샤아트홀	20180424	20180504	0원

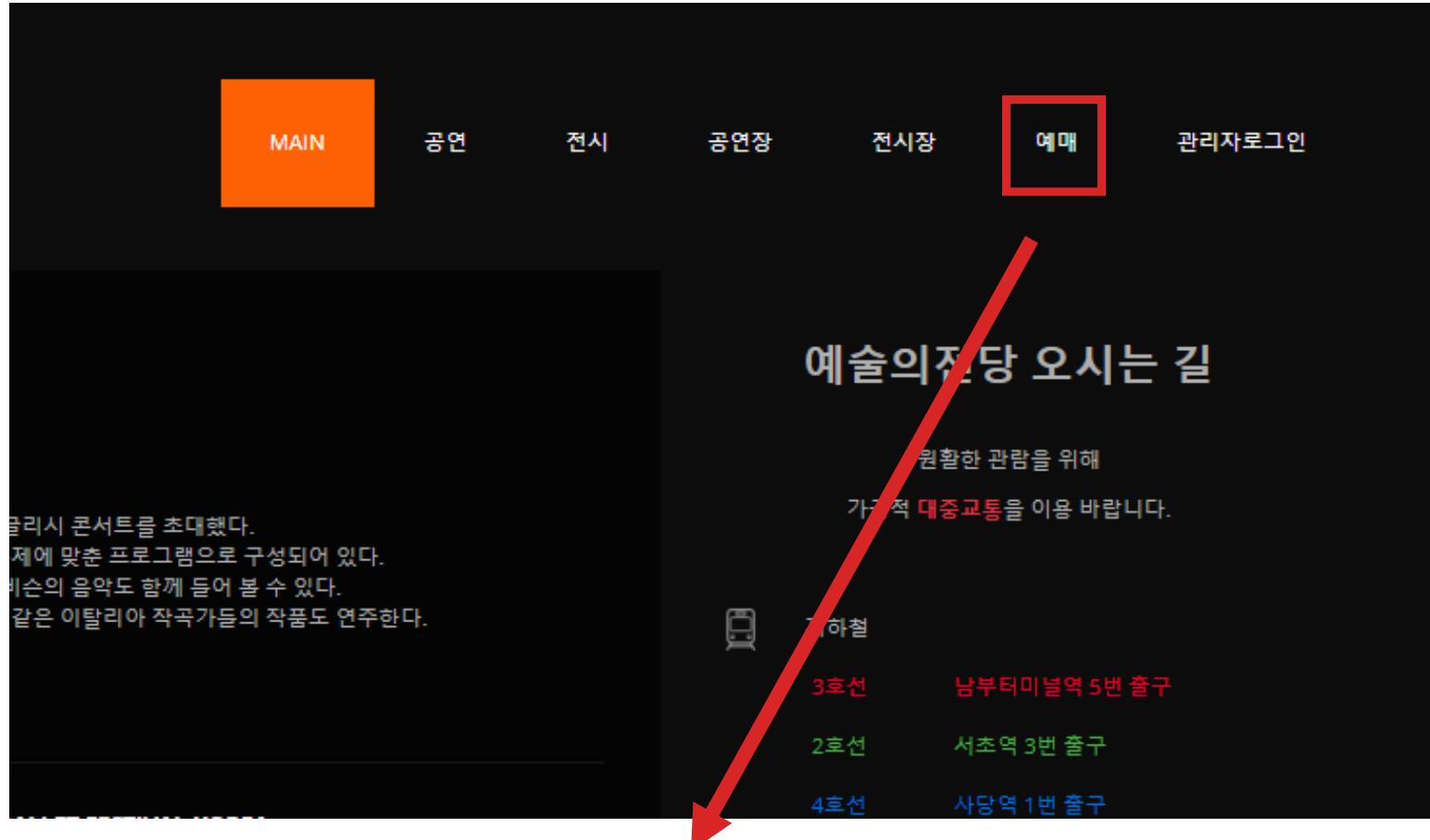
```
ArtDB.makeDatabase();
ResultSet rs = ArtDB.outtrs("select * from perform");
try{
    while(rs.next()){ %>
<tr>
    <th scope="row"><a href="perpick.html"><%=rs.getString("pername")%></a></th>
    <td><%=rs.getString("perfullname")%></td>
    <td><%=rs.getString("perfrom")%></td>
    <td><%=rs.getString("perto")%></td>
    <td><%=rs.getString("pertime")%></td>
    <td><%=rs.getString("perprice")%>원</td>
    <td><%=rs.getString("perseat")%>석</td>
</tr>
```

```
public static ResultSet outtrs(String string) {
    try {
        rs = stmt.executeQuery(string);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return rs;
}
```

- **outtrs** 메소드를 통해

**Database에 질의 요청을 하는 방식**

# 로그인 페이지



아트 모아 로그인

로그인 회원가입

ID  
Password

로그인



- 비회원 예매는 제공하지 않기 때문에  
예매 버튼을 클릭하게 되면 로그인 페이지로 이동한다.

- 로그인 시, 데이터베이스의 ID, PW값과 대조하여  
일치하지 않는 경우, 로그인을 할 수 없다.

# 회원가입 구현

아트모아로그인

로그인 회원가입

Terms and conditions



- 회원가입 시, 양식 입력 후 회원가입 버튼을 누르면  
명령이 수행되어 DB에 저장된다

```
ic static void makeDatabase() {  
// 드라이버 로딩  
try {  
    Class.forName("org.gjt.mm.mysql.Driver");  
try {  
    // 연결하기 - artmoa 데이터베이스와 연결  
    String URL = "jdbc:mysql://localhost:3306/artmoa?useUnicode=true&characterEncoding=UTF-8";  
    con = DriverManager.getConnection(URL, user: "root", password: "onlyroot");  
    System.out.println("\n" + URL + "에 연결됨");  
    stmt = con.createStatement();
```

## Step. 1

- **makeDatabase** 메소드로 데이터베이스와 연결  
(추후 모든 작업의 첫 번째 과정으로 쓰인다)

```
if (pw.length() >= 8) {  
    out.print("<script>alert('입력하신 PW는 8자 이상입니다.')</script>");  
    out.print("<script>window.top.location='login.html'</script>");  
}
```

## Step. 2

- 최소 PW 수(8자리)를 체크한다
- 8자리 이상이면 alert 메시지를 띄우며 회원가입이 제한된다

# 회원가입 구현

```
public static int findMember(String memid) {  
    int cnt=0;  
    try {  
        // SQL 질의문을 수행한다.  
        String sql = "select * from member where memid=?";  
        //System.out.println("      >> SQL : " + sql + "\n");  
        prstmt = con.prepareStatement(sql);  
        prstmt.setString( parameterIndex: 1, memid);  
        rs = prstmt.executeQuery();  
        if (rs.next()) {  
            cnt++;  
        }  
    }
```

```
ArtDB.makeDatabase();  
  
String id = request.getParameter("memid");  
String pw = request.getParameter("mempw");  
String name = request.getParameter("memname");  
name = new String(name.getBytes("8859_1"), "UTF-8");  
String tel = request.getParameter("memtel");  
  
int result = ArtDB.insertMember(id, pw, name, tel);
```

## Step. 3

- findMember 메소드
- SQL 질의를 통하여 중복된 ID가 있는지 검사한다

## Step. 4

- 중복된 아이디가 없다면, insertMember 메소드를 이용하여 입력한 데이터를 DB에 삽입한다

# 예매 가능

ArtMOA

MAIN 공연 전시 공연장 전시장 예매 관리자로그인

예매하실 컨텐츠를 선택해주세요.

전시 예매 | 공연 예매 | 전시 예매 확인 | 공연 예매 확인 | 전시 예매 취소 | 공연 예매 취소

제목	장소	시작날짜	끝날짜	시간	가격	좌석수
2018 교향악축제	콘서트홀	20180430	20180430	오후7시30분	30000원	2527석
2018 장선희발레단<러브스토리>	CJ토월극장	20180526	20180527	오후2시, 오후5시	50000원	1004석
KCO 트로이카콘서트	IBK챔버홀	20180415	20180415	오후3시, 오후7시	30000원	600석
누오바오페라단 여우던	오페라극장	20180511	20180513	오후6시	120000원	2283석
신과함께 저승편	CJ토월극장	20180327	20180415	오후2시, 오후7시	60000원	1004석
연극<비람 불어 별이 흔들릴 때>	리사이틀홀	20180418	20180506	오후6시30분	30000원	354석

공연명: 2018 교향악축제  
관람날짜(예:20180606) : null  
관람시간(예:오후2시) : null  
관람인원 : 1

예매하기 | 취소

관람날짜(예:20180606) : <input type="text" name="date" size="10" value="20180606" />  
관람시간(예:오후2시) : <input type="text" name="time" size="10" value="1400" />  
관람인원 : <select name="people">



```
String id = (String)session.getAttribute("log_id");
String pname = request.getParameter("name");
pname = new String(pname.getBytes("8859_1"), "UTF-8");
String pdate = request.getParameter("date");
String ptime = request.getParameter("time");
ptime = new String(ptime.getBytes("8859_1"), "UTF-8");
int pseatnum = Integer.parseInt(request.getParameter("people"));
```

- 전시, 공연 예매 가능
- 회원이 예매한 내역 확인 가능

- 회원이 입력한 정보를 html -> java 변수로 저장하는 과정

# 예매 기능

```
if(pdate.equals("null")){
    out.print("<script>alert('공연 예매에 실패하였습니다.');//</script>");
    out.print("<script>window.top.location='resevper.jsp'</script>");
}
else if(ptime.equals("null")){
    out.print("<script>alert('공연 예매가 실패하였습니다.');//</script>");
    out.print("<script>window.top.location='resevper.jsp'</script>");
}
```

```
int result = ArtDB.perreserve(id, pname, pdate, ptime, pseatnum);

out.print("<script>alert('공연 예매가 완료되었습니다.');//</script>");
out.print("<script>window.top.location='resevper.jsp'</script>");
```

- 예매 버튼을 눌렀을 경우,

누락되어 있는 정보(date, time)가 있다면 '예매 실패'

- 그렇지 않다면 perreserve 메소드를 통해 예매 처리

```
public static int perreserve(String memid, String pername, String Pdate, String Ptime, String Pseatnum) {
    int mod = 0;
    int Pnum = rowCount(table: "per_reserve") + 1;

    try {
        mod = stmt.executeUpdate(sql: "insert into per_reserve values('" + Pname + "','" + Pdate + "','" + Ptime + "','" + Pseatnum + "')");
    } catch (SQLException e) {
        // 예외 처리
    }
}
```

- DB 연결 후 insert 문을 통하여 데이터 삽입



- 이후 예매현황 페이지로 이동되어

회원이 예매 한 내역을 확인할 수 있음

# 예매 취소 기능

예매를 취소하실 공연 예매 번호를 선택해주세요

전시 예매	공연 예매	전시 예매 확인	공연 예매 확인	전시 예매 취소	공연 예매 취소
-------	-------	----------	----------	----------	----------

예매자 아이디	예매 번호	예매 공연명	관람 날짜
ME100	4	신과함께 저승편	20160612

취소할 공연 번호: 4 ▼

예매 취소	다시 고르기
-------	--------

```
ResultSet rs2 = ArtDB.checkReserve_per(id);
try{
    while(rs2.next()){
        <option value="<%="rs2.getString("Pnum")%>"><%=rs2.getString("Pnum")%>
    }
}
```

- 전시회나 공연 일정에 고유 ID 값을 부여

Pnum	memid	pername	Pdate	Ptime	Pseatnum
1	ME400	신과함께 저승편	20180415	오후2시	120
2	ME500	두오바오페라단 여우연	20180511	오후6시	125
3	ME600	KCO 트로이카콘서트	20180415	오후7시	130

```
public static int percancel(int Pnum, String memid) {
    int mod = 0;

    try {
        mod = stmt.executeUpdate( sql: "delete from per_reserve Where Pnum = "
            + Pnum + " and memid = '" + memid + "'");
    } catch (SQLException e) {
        System.err.println("\n ---- 입력하신 예매번호의 예매내역이 해당 아이디와 일치하지 않습니다.\n");
        e.printStackTrace();
    }
}
```

- 예매 취소 시, 고유 번호와 예매 했던 회원의 ID가 모두 일치해야 예매 취소 가능

localhost:8080 내용:

선택하신 공연 예매가 취소되었습니다.

확인

# Admin Login

ADMIN ONLY

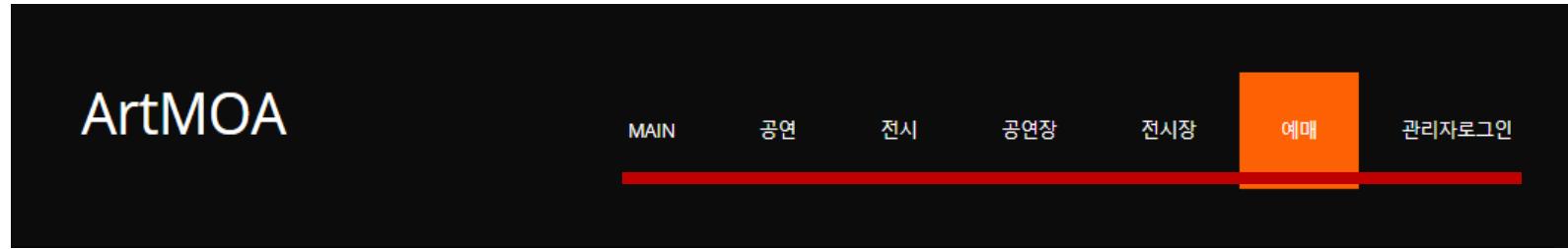


-ArtMOA ADMIN Only Page-

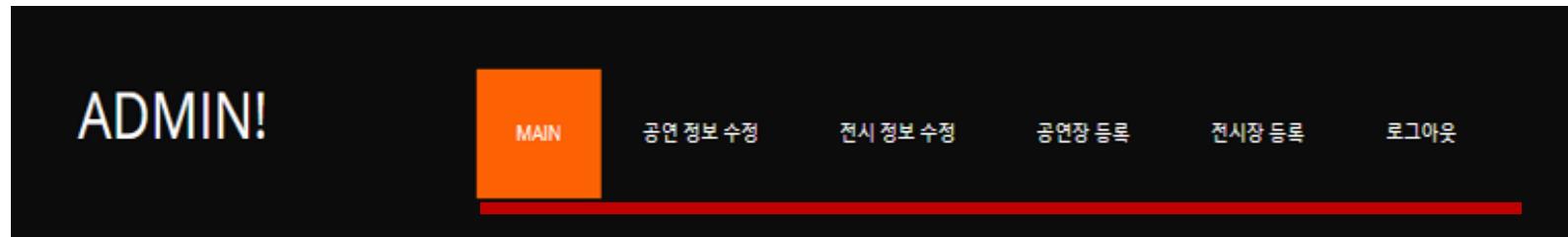
adminid	adminpw	adminname	admintel
AD100	1111	나수영	01011111111
AD200	2222	이찬수	01022222222
AD300	3333	정기태	01033333333

- 관리자 DB에 저장되어 있는 값으로만 로그인 가능

# Admin Mode



- 비회원 및 회원이 볼 수 있는 메뉴

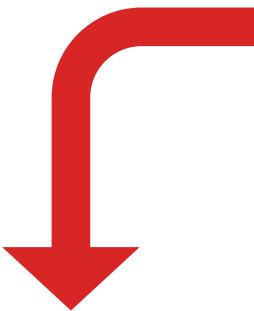


- 관리자 ID/PW로 로그인해야만 볼 수 있는 메뉴

## 관리자 기능

- 전시장과 공연장 등록 및 삭제 가능
- 전시회, 공연 일정 등록 및 삭제 가능

# Admin Mode(등록)



2018 장선희발레단<러브스토리> CJ托월극장

공연 정보 등록하기

발레공연

공연 장소: CJ托월극장

20180612

20180630

오후2시|오후4시

50000

좌석 수: 200

```
<h1><center>공연 정보 등록하기</center></h1>
<input type="text" name="name" id="name" placeholder="공연 이름" required/><br><br>
공연 장소 : <select name="hall">
    <%
        ResultSet rs2 = ArtDB.outtrs("select perhallname from per_hall");
        try{
            while(rs2.next()){ %>
                <option value="<%=rs2.getString("perhallname")%>"><%=rs2.getString("perhallname")%>
            }
        }catch (SQLException e) {
            e.printStackTrace();
        }%>
    </select><br><br>
<input type="text" name="from" id="name" placeholder="시작 날짜(예:20180606)" required/>
<input type="text" name="to" id="name" placeholder="끝 날짜(예:20180606)" required/>
<input type="text" name="time" id="name" placeholder="공연 시간(예:오후2시)" required/>
<input type="text" name="price" id="name" placeholder="가격(예:10000)" required/><br><br>
좌석 수 : <select name="seat">
```

```
String name = request.getParameter("name");
name = new String(name.getBytes("8859_1"), "UTF-8");
String hall = request.getParameter("hall");
hall = new String(hall.getBytes("8859_1"), "UTF-8");
String from = request.getParameter("from");
String to = request.getParameter("to");
String time = request.getParameter("time");
time = new String(time.getBytes("8859_1"), "UTF-8");
int price = Integer.parseInt(request.getParameter("price"));
int seat = Integer.parseInt(request.getParameter("seat"));

int result = ArtDB.modPer1(name, hall, from, to, time, price, seat);
```

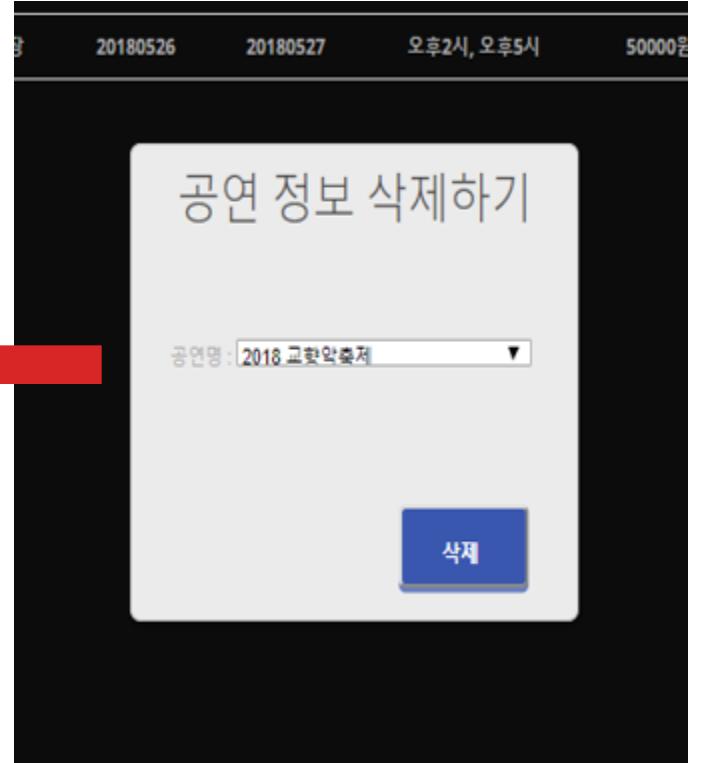
```
public static int modPer1(String pername, String perhallname, String perfrom, String perto, String pertime,
                           int perprice, int perseat) {
    int mod = 0;

    try {
        mod = stmt.executeUpdate(sql: "insert into perform values('"
            + pername + "','" + perhallname + "','" + perfrom
            + "','" + perto + "','" + pertime + "','" + perprice + "," + perseat + ")");
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return mod;
}
```

- 관리자가 입력한 값을 DB에 저장하여 등록 처리

# Admin Mode(삭제)



```
<form action="del_perinfo.jsp">
<h1><center>공연 정보 삭제하기</center></h1>
<br><br><br>
    공연명 : <select name="name">
        <%
            ResultSet rs3 = ArtDB.outtrs("select pername from perform");
            try{
                while(rs3.next()){ %>
                    <option value=<%=rs3.getString("pername")%>><%=rs3.getString("pername")%>
                }
            }
        <%>
    </select>
</form>
```

```
ArtDB.makeDatabase();

String name = request.getParameter("name");
name = new String(name.getBytes("8859_1"), "UTF-8");

int result = ArtDB.delPerform(name);

if(result > 0){
    out.print("<script>alert('공연 일정이 삭제되었습니다.');//</script>");
    out.print("<script>window.top.location='perinfomod.jsp'</script>");
}

public static int delPerform(String pername) {
    int mod = 0;

    try {
        String sql = "delete from perform where pername = ?;";
        //System.out.println("    >> SQL : " + sql + "\n");
        prStmt = con.prepareStatement(sql);

        prStmt.setString( parameterIndex: 1, pername);

        mod = prStmt.executeUpdate();
    }
}
```

- 선택한 이름의 정보를 DELETE 문으로 삭제 처리

## 2ND PROJECT

# 뷰로 만든 유튜브 "VueTube"

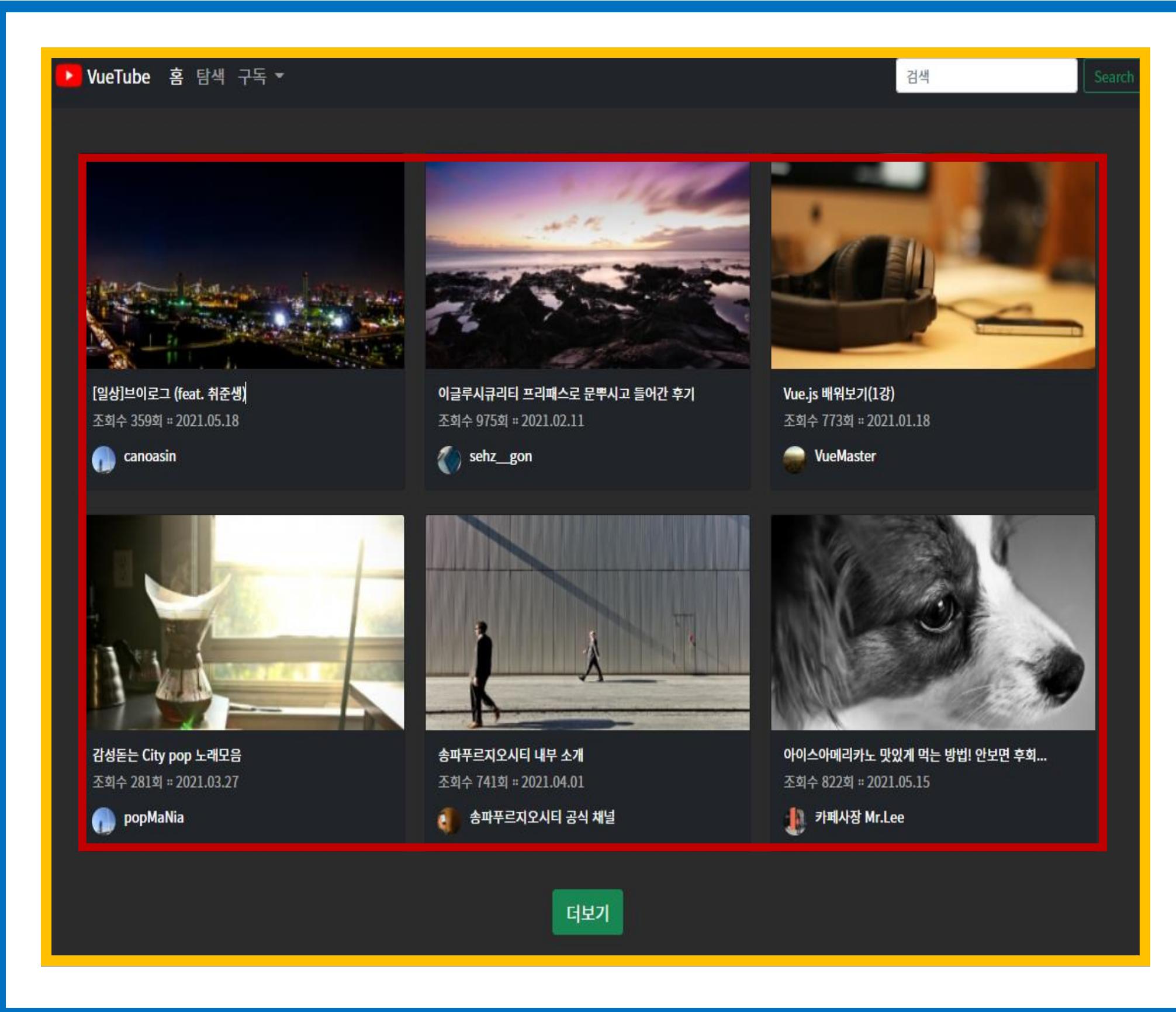
### 소개 및 기능

- YouTube의 Clone Site를 구현하여, 유튜브에서 제공하는 기능을 일부 구현
- 여러 개의 템플릿을 컴포넌트화 하고, Vue Router를 사용하여 관리 및 유지보수 용이
- axios를 활용하여 외부 데이터를 불러올 수 있음.

### 개발 환경 및 사용 기술

- Visual Studio Code
- Vue.js
- Javascript
- html/css
- Bootstrap

# 메인화면 / 구조



## App.vue

- 부모 컴포넌트

## Home.vue

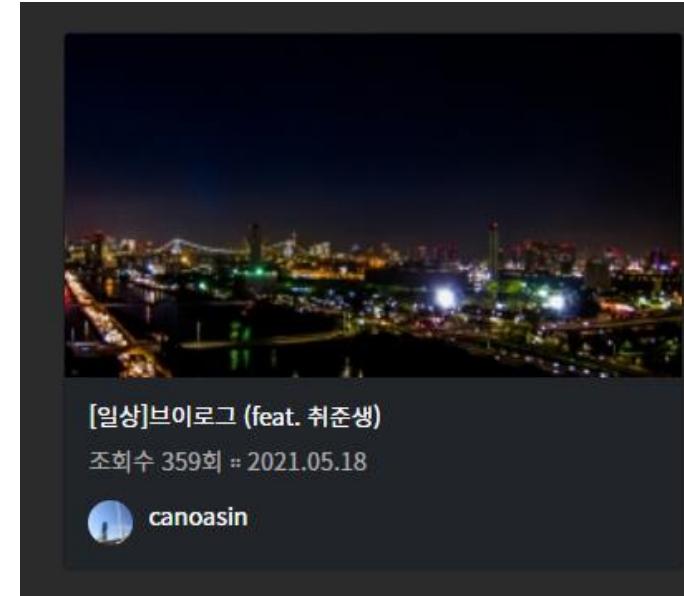
- App.vue의 하위 컴포넌트로, 메인 화면을 담당

## Card.vue

- Home.vue의 하위 컴포넌트로, 각 영상들의 디자인 및 데이터 출력을 담당

# 메인화면 / 구조

```
id : 0,  
title: "[일상]브이로그 (feat.  
image: "https://placeimg.com/  
userImage: "https://placeimg.com/  
username : "canoasin",  
date: "2021.05.18",  
click : 359,  
likes : 53,  
hates : 12,  
subscribe : 50
```



- data.js에 저장되어 있는 데이터를 props를 사용하여 데이터바인딩하고 컴포넌트화 하여 구현

```
<script>  
import 데이터 from './assets/data.js';
```

```
export default {  
  name: 'App',  
  data(){  
    return {  
      영상들 : 데이터,
```

- data.js를 "영상들" 이라는 이름으로 지정하여 사용.

```
<script>  
import Card from './Card.vue';  
  
props : {  
  영상들 : Object,  
},  
  
components: {  
  Card : Card  
},
```



```
<div class="row row-cols-1 row-cols-md-3">  
  <Card :영상들="영상들[i]" v-for="(a,i) in 영상들" :key="i"/>  
</div>
```

- Home.vue에서 props로 데이터를 받아 Card라는 이름의 컴포넌트를 정의, 사용하는 과정

```
<template>  
  <div class="col content-pad">  
    <div class="card bg-dark h-100">  
      <a v-bind:href="`/detail/${영상들.id}`"></a>  
      <div class="card-body">  
        <a v-bind:href="`/detail/${영상들.id}`"><h6 class="card-title">{{ 영상들.title }}</h6></a>  
        <p class="card-text muted">조회수 {{ 영상들.click }}회 :: {{ 영상들.date }}</p>  
        <div class="profile" :style="{ backgroundImage : 'url('+영상들.userImage+')' }"></div>  
        <h6 class="card-text"> {{ 영상들.username }}</h6>  
      </div>  
    </div>  
  </div>
```

- Card.vue

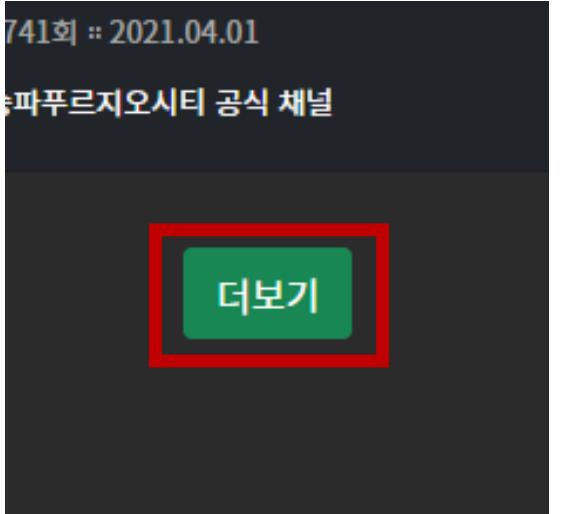
# Vue Router 구성

```
const routes = [  
  {  
    path: "/",  
    component: Home,  
  },  
  {  
    /* path: "/detail/:id", */  
    path: "/detail/:id",  
    component: Detail,  
  },  
  {  
    path: "/:anything(.*)",  
    component: ErrorPage,  
  },  
];
```

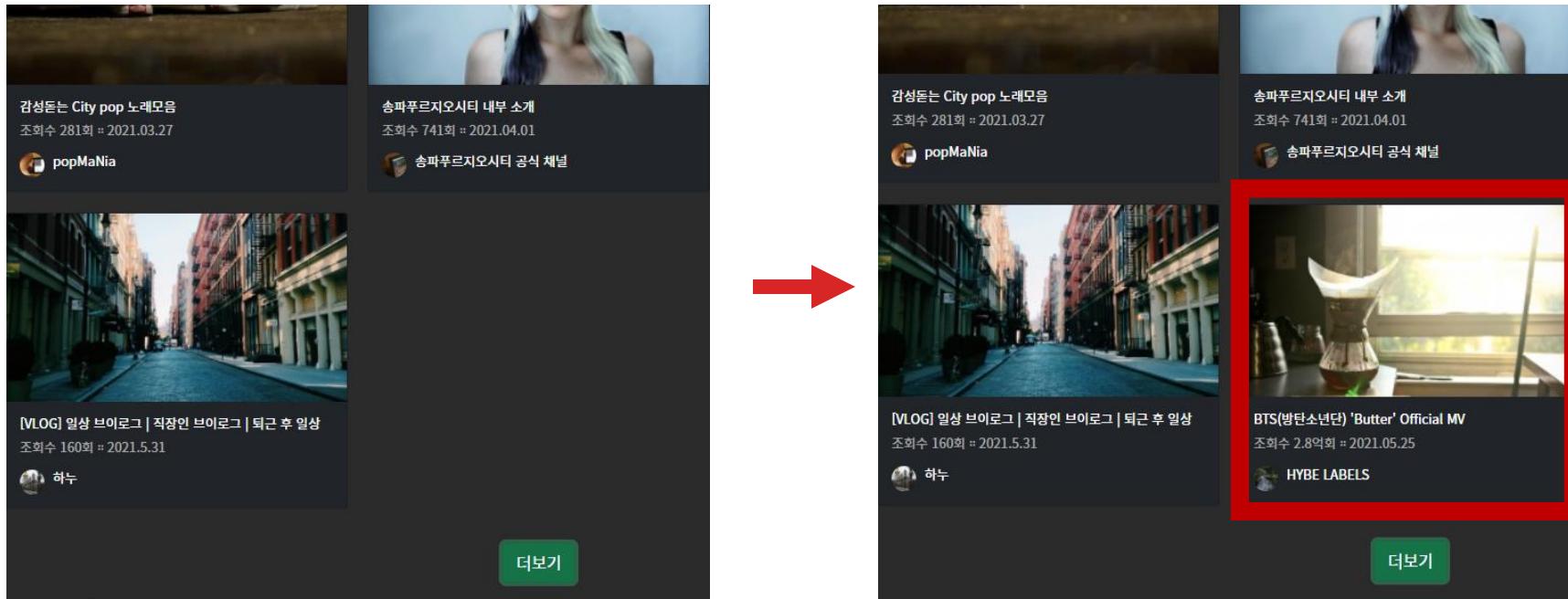
- 1 - 주소에 아무것도 입력하지 않았을 때  
나타나는 메인 페이지(Home 컴포넌트)
- 2 - 영상을 클릭했을 때 나타나는 디테일 페이지  
(Detail 컴포넌트)
- 3 - 유효하지 않은 주소로 접근했을 때 나타나는  
404 Error 페이지 (ErrorPage 컴포넌트)



# axios를 활용한 더보기 버튼



- 메인 페이지 하단의 '더보기' 버튼을 누르면 axios로 가져온 데이터가 생성된다.



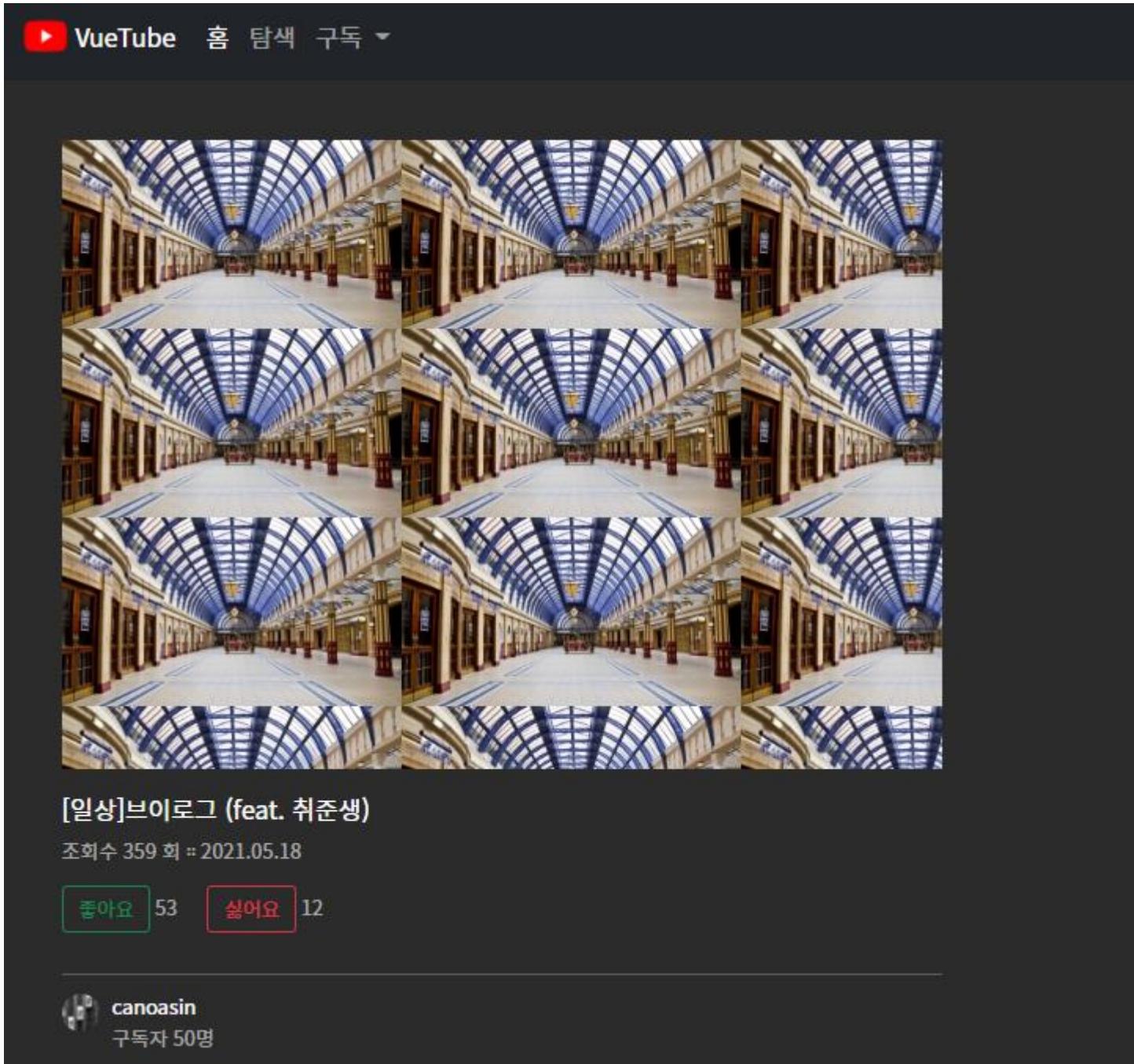
```
methods : {
  more() {
    axios.get(`https://raw.githubusercontent.com/canoasin/vue/main/more${this.더보기}.json`)
      .then( 결과 => {
        this.영상들.push(결과.data);
        this.더보기++;
      })
  }
},
```

- 더보기 기능을 제공하는 'more' 메소드

```
{
  "id" : 6,
  "title": "[VLOG] 일상 브이로그 | 직장인 브이로그 | 퇴근 후 일상",
  "image": "https://placeimg.com/270/150/arch",
  "userImage": "https://placeimg.com/100/100/arch",
  "username" : "하누",
  "date": "2021.5.31",
  "click" : 160,
  "likes" : 50,
  "hates" : 3,
  "subscribe" : 469
}
```

- axios를 통해 깃허브에 올린 array 데이터를 기존 데이터(영상들)에 push한다.
- 파일명을 more0, more1, more2 …로 작명하여 더보기 버튼을 누를 때마다 1씩 증가하도록 구현했다

# Detail 페이지



- 각 영상 객체를 클릭하면 detail 페이지로 연결된다.

```
// class="full" :style="{ backgroundImage: 'url('+videoList[$route.params.id].image+')' }" class="muted">> 조회수 {{ videoList[$route.params.id].viewCount }}
```

- Array마다 id 번호를 하나씩 부여
- \$route.params를 이용하여 해당 id값의 array를 출력하도록 구현

```
id : 0  
title: "[일상] 브이로그 (feat. 취준생)"  
image: "https://.../  
userImage: "http://.../  
username : "canoasin"  
date: "2021.05.18"
```

```
path: "/detail/:id",  
component: Detail,
```

localhost:8080/detail/0

# 좋아요/싫어요/구독 가능



```
data() {
  return {
    likes : '',
    hates : '',
}
created() {
  this.likes = this.영상들[this.$route.params.id].likes
  this.hates = this.영상들[this.$route.params.id].hates
},
```

- 따라서 created 사이클에 해당 id값을 가진 array의 좋아요/싫어요 개수를 저장하여 사용했다.

- 좋아요/싫어요 기능을 하는 함수를 만들고

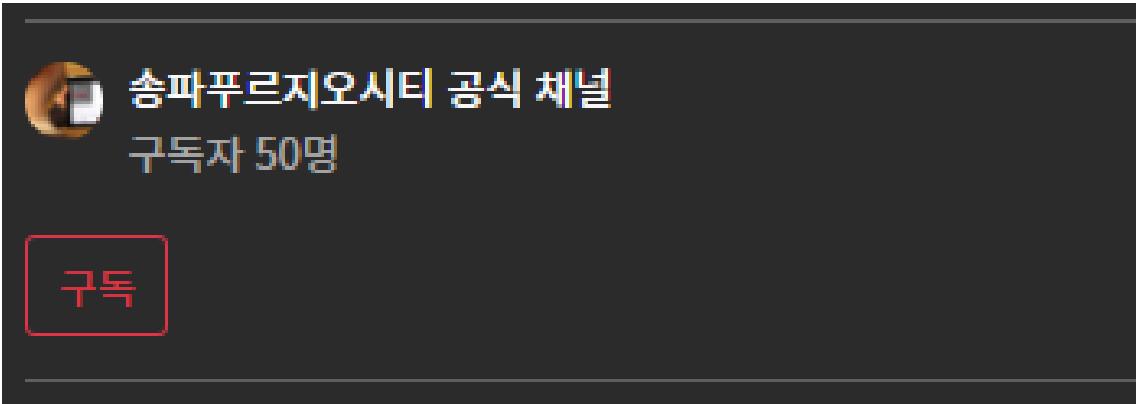
버튼을 누를 때 실행함

- \$route.params.id 값으로 array를 불러왔기 때문에 함수를 만드려면  
해당 array의 좋아요/싫어요 개수를 저장하는  
변수가 필요했다

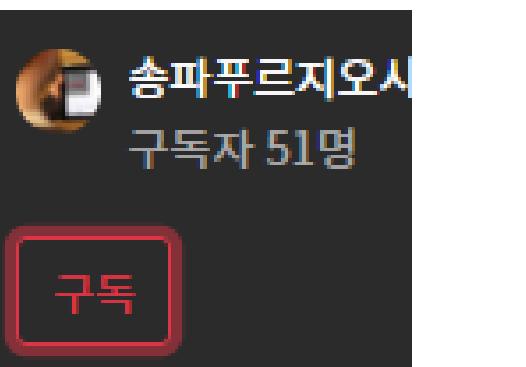
```
like () {
  if (this.ifLike == false) {
    this.likes++;
    this.ifLike = true;
  } else if (this.ifLike == true) {
    this.likes--;
    this.ifLike = false;
  }
},
hate () {
  if (this.ifHate == false) {
    this.hates++;
    this.ifHate = true;
  } else if (this.ifHate == true) {
    this.hates--;
    this.ifHate = false;
  }
}
```

- 버튼이 눌러있지 않을 때 1 증가
- 버튼이 눌러있을 때 1 감소

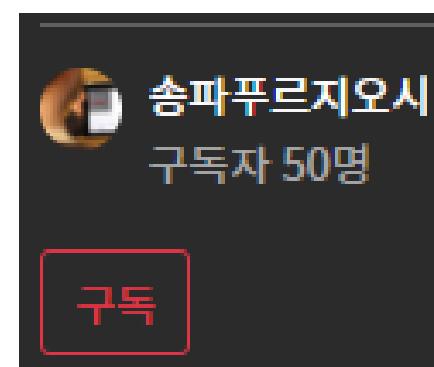
# 좋아요/싫어요/구독 기능



- 구독 기능도 마찬가지로 버튼을 누르면 실행



- 구독버튼 클릭



- 구독버튼 재클릭

```
ifsub : false,  
subs : '',
```

- 변수 생성 및 초기화

```
this.subs = this.영상들[this.$route.params.id].subscribe
```

- created 사이클 내부 subs 변수에 데이터값 저장

```
subscribe () {  
  if (this.ifsub == false) {  
    this.subs++;  
    this.ifsub = true;  
  } else if (this.ifsub == true) {  
    this.subs--;  
    this.ifsub = false;  
  }  
},
```

- 버튼이 눌러있지 않을 때 1 증가

- 버튼이 눌러있을 때 1 감소

감사합니다.