WGWG

공연정보 기록 웹사이트

송나은

https://github.com/ssongbt/wg.git

WGWG

CONTENTS

프로젝트 소개

작업 기간

구현하고자 한 내용

시연 영상

프로젝트 주제

WGWG

WGWG(와글와글)
공연 정보를 기록할 수 있는 웹사이트
자신이 본 공연에 대한 정보를 기록하고 눈으로 확인 할 수 있는 사이트

주요 기능

공연정보 검색

관람 공연,
관람 예정 공연,
관람 희망 공연
등록

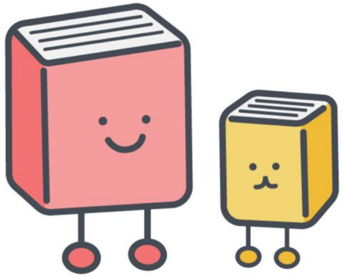
관람 공연
리뷰 작성

관람한
정보 데이터
확인

작업 기간

2022.04.05 - 04.28

참고 사이트



도서 기록 앱
〈북적북적〉

PLAY DB

공연 정보 사이트
〈플레이 DB〉

개발환경

spring

eclipse

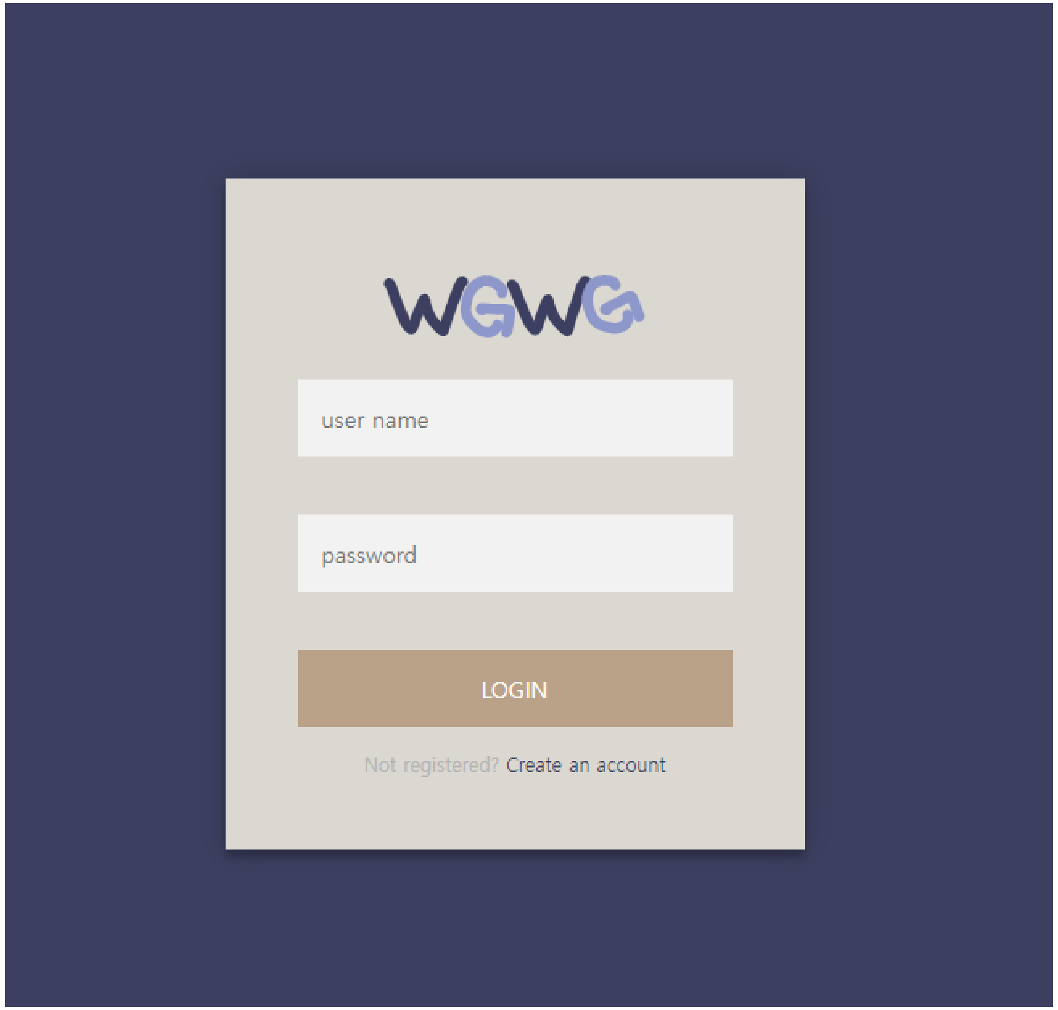
MySQL

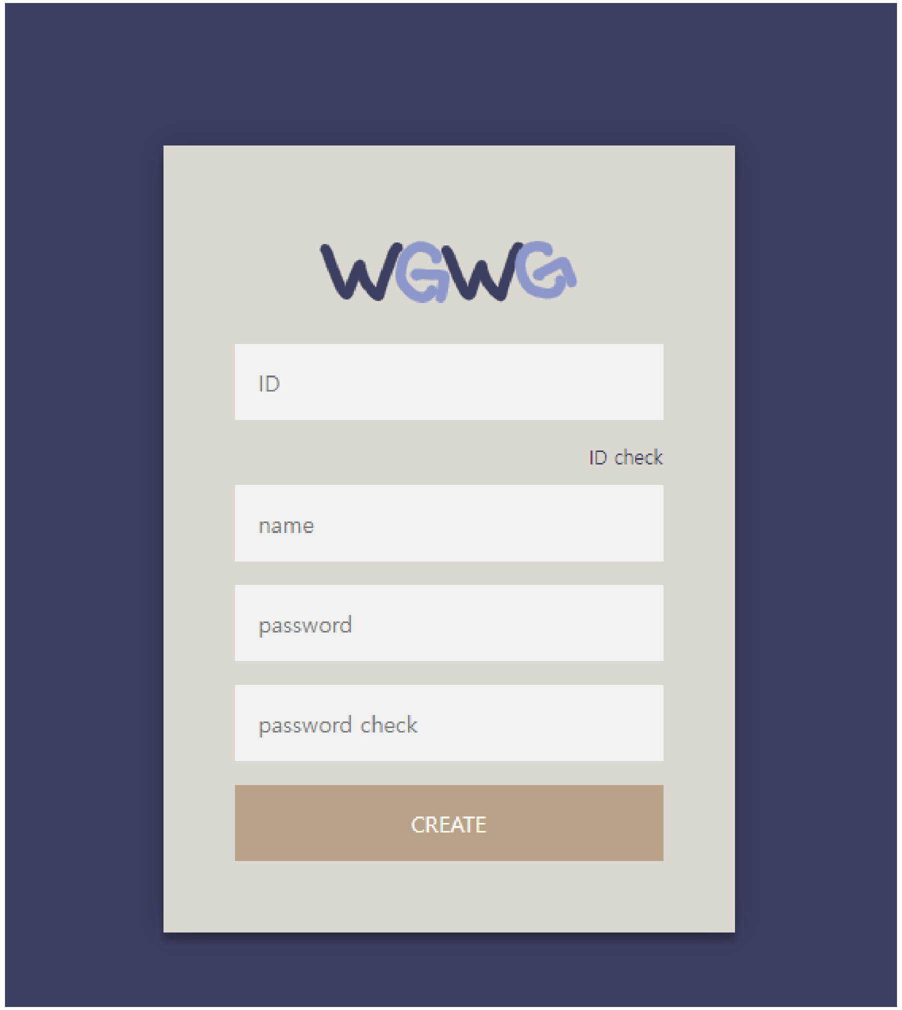
Windows

구현하고자 한 내용

1. 회원 가입
2. 공연 검색 기능(배우, 극 이름, 극장이름)
3. 공연 정보 제공
4. 공연을 리스트 형태로 제공
5. 자신이 본 공연 저장/수정/삭제
6. 자신이 볼 예정인 공연 저장/수정/삭제
7. 자신이 보고 싶은 공연 저장/수정/삭제
8. 공연 정보를 누적하여 차트로 제공
9. 자신이 본 공연을 리스트 형태로 제공

1. 회원가입 (+ 로그인)


WGWG
user name
password
LOGIN
Not registered? Create an account


WGWG
ID
ID check
name
password
password check
CREATE

1. 회원가입 (+ 로그인)

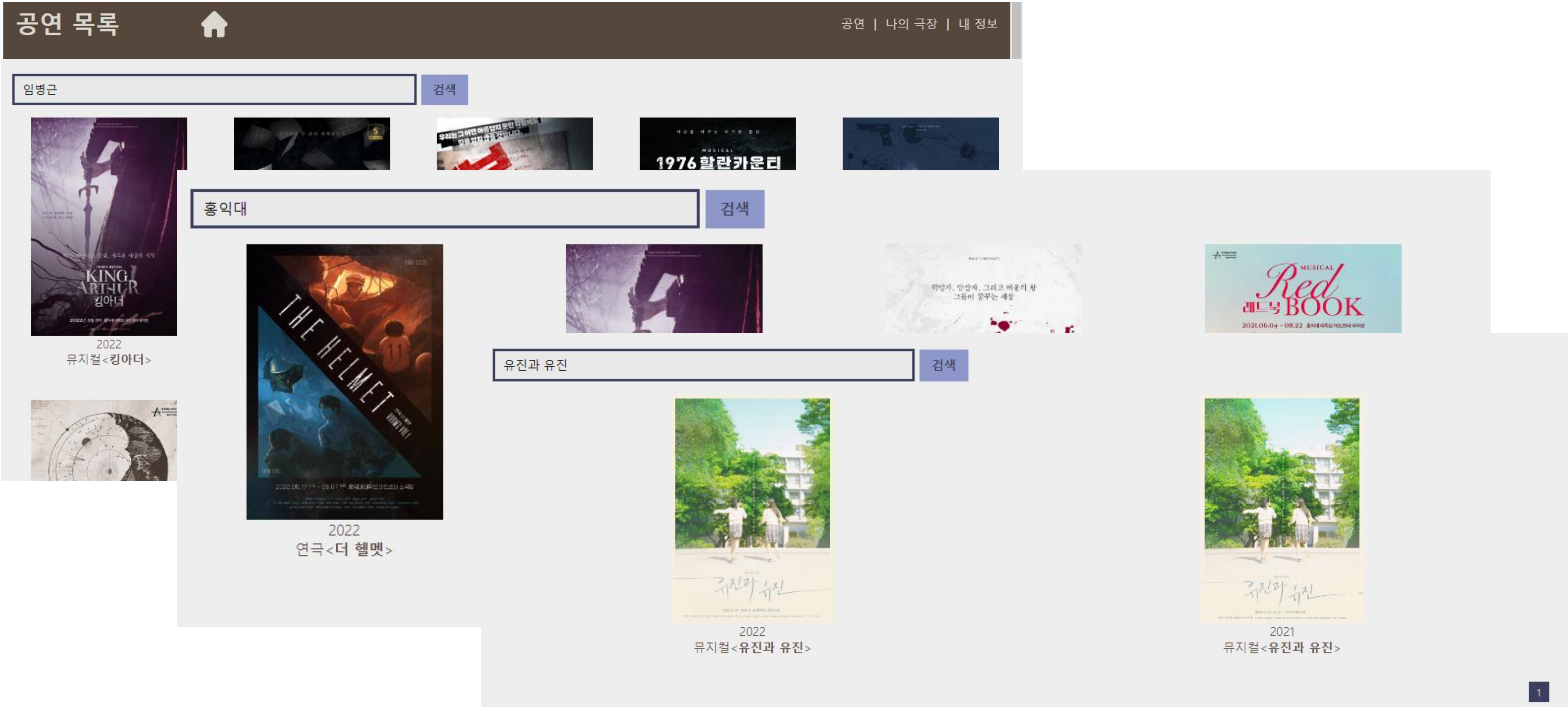
Controller

```
@GetMapping("/login")
public String form() {
return "pj/loginForm";
}

@PostMapping("/login")
@ResponseBody
public Map<String, Object> login(WgUserVO user, Model model) {
Map<String, Object> map = new HashMap<>();
boolean ok = svc.login(user);
if(ok) {
model.addAttribute("uid", user.getUid());
}
map.put("ok", ok);
return map;
}

@GetMapping("/join")
public String joinform() {
return "pj/joinForm";
}

@PostMapping("/join")
@ResponseBody
public Map<String, Boolean> join(WgUserVO user) {
boolean useradded = svc.addUser(user);
Map<String, Boolean> map = new HashMap<>();
map.put("useradded", useradded);
return map;
}

@PostMapping("/idcheck")
@ResponseBody
public Map<String, Boolean> check(@RequestParam String uid) {
Map<String, Boolean> map = new HashMap<>();
boolean checked = svc.idcheck(uid);
map.put("checked", checked);
return map;
}
```

Mapper

```
<insert id="addUser"
parameterType="com.tjoeun.vo.WgUserVO">
INSERT INTO wgusers VALUES(#{uid}, #{pwd}, #{nick})
</insert>

<select id="getUserById"
resultType="com.tjoeun.vo.WgUserVO"
parameterType="String">
SELECT uid,pwd,nick
        FROM wgusers
        WHERE uid=#{uid}
</select>

<select id="login"
    resultType="String"
    parameterType="com.tjoeun.vo.WgUserVO">
    SELECT uid
    FROM wgusers
    WHERE uid=#{uid} AND pwd=#{pwd}
</select>

<update id="updateUser"
parameterType="com.tjoeun.vo.WgUserVO">
UPDATE wgusers
SET pwd=#{pwd}, nick=#{nick}
WHERE uid=#{uid}
</update>

<delete id="deleteUser" parameterType="String">
DELETE FROM wgusers WHERE uid=#{uid}
</delete>
```

2. 공연 검색 기능(배우, 극 이름, 극장이름)



공연 목록

공연 | 나의 극장 | 내 정보

임병근 검색

2022
뮤지컬<킹아더>

홍익대 검색

2022
연극<더 헬멧>

유진과 유진 검색

2022
뮤지컬<유진과 유진>

2021
뮤지컬<유진과 유진>

1

2. 공연 검색 기능(배우, 극 이름, 극장이름)

Controller

```
@GetMapping("/list")
//@ResponseBody
public String getPageSearch(@SessionAttribute(name="uid",
required=false) String uid, @RequestParam(required=false,
defaultValue="1") int pageNum, Model model,
@RequestParam(defaultValue="") String keyword) {
      PageInfo<Map<String, Object>> pageInfo =
svc.getListPageSearch(pageNum, keyword);
      pageInfo.getNavigatepageNums();
model.addAttribute("pageInfo",pageInfo);
      model.addAttribute("list",pageInfo.getList());
      model.addAttribute("keyword",keyword);
      model.addAttribute("uid",uid);
      return "pj/listPageSearch2";
}
```

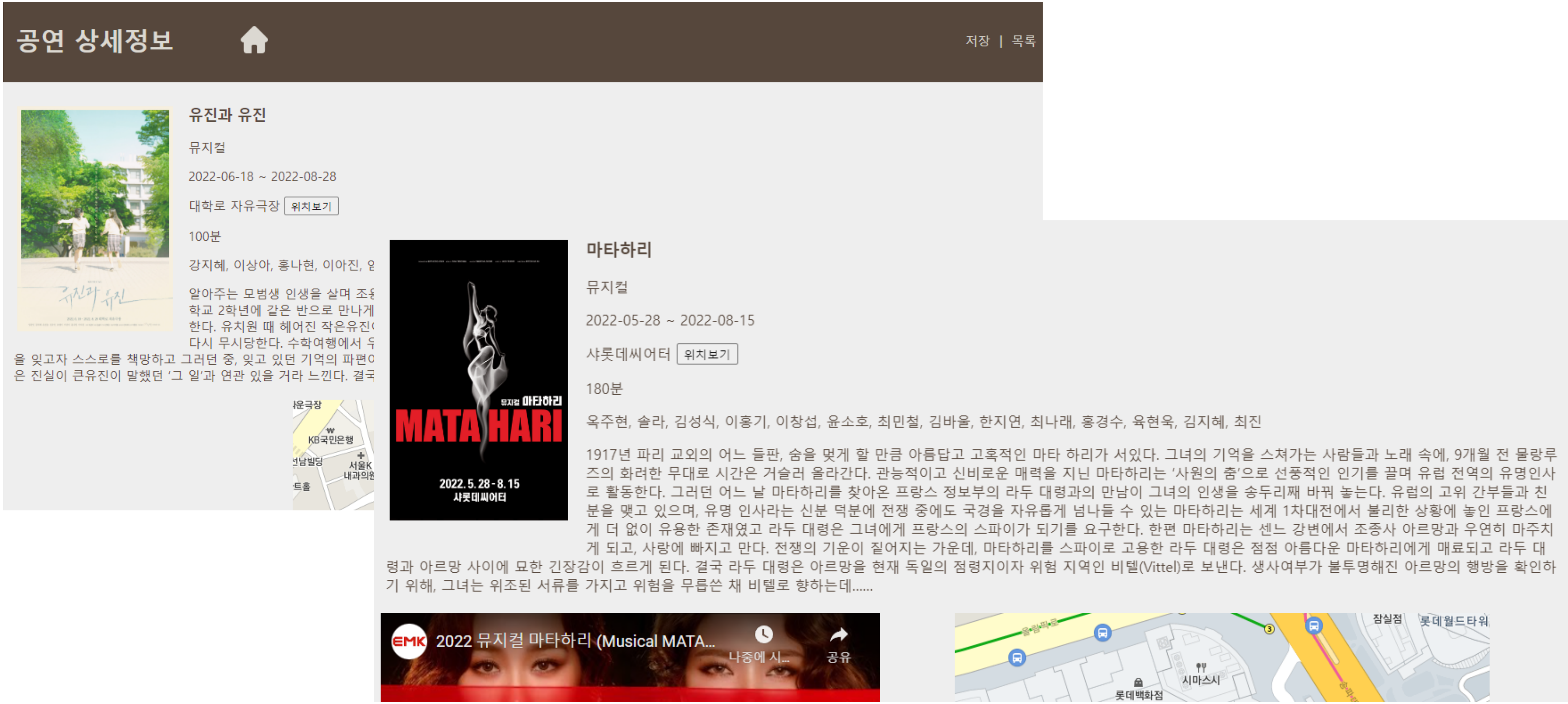
DAO

```
public PageInfo<Map<String, Object>> getListPageSearch(int pageNum,
String keyword) {
      PageHelper.startPage(pageNum, 10);
      PageInfo<Map<String, Object>> pageInfo = new
PageInfo<>(dao.searchPerform(keyword));
      return pageInfo;
}
```

Mapper

```
<select id="performanceList"
resultType="map">
SELECT * FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
ORDER BY p.startdate DESC
</select>

<select id="searchPerform"
resultType="map"
parameterType="String">
SELECT * FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE name LIKE CONCAT('%',#{keyword},'%') OR loc LIKE CONCAT('%',#{keyword},'%') OR cast LIKE CONCAT('%',#{keyword},'%')
ORDER BY p.startdate DESC
</select>

<select id="getTotal"
resultType="int">
SELECT COUNT(*) FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE name LIKE CONCAT('%',#{keyword},'%') OR loc LIKE CONCAT('%',#{keyword},'%') OR cast LIKE CONCAT('%',#{keyword},'%')
</select>

<insert id="addPerform"
parameterType="com.tjoeun.vo.WgPerformVO"
useGeneratedKeys="true" keyProperty="num">
INSERT INTO wgperformance VALUES(null, #{name}, #{startdate}, #{enddate}, #{loc}, #{genre}, #{run}, #{cast}, #{synop})
</insert>

<insert id="addImage"
parameterType="com.tjoeun.vo.WgImageVO"
useGeneratedKeys="true" keyProperty="img_num">
INSERT INTO wgperformimage VALUES(null, #{perform_num}, #{imagepath})
</insert>

<update id="updatePerform"
       parameterType="com.tjoeun.vo.WgPerformVO">
       UPDATE wgperformance
       SET name=#{name}, startdate=#{startdate}, enddate=#{enddate}, loc=#{loc}, genre=#{genre}, run=#{run}, cast=#{cast}, synop=#{synop}
       WHERE num=#{num}
    </update>

    <update id="updateImage"
    parameterType="com.tjoeun.vo.WgImageVO">
    UPDATE wgperformimage
    SET imagepath=#{imagepath}
    WHERE perform_num=#{num}
    </update>

<select id="getFilename"
parameterType="Integer"
resultType="String">
SELECT imagepath
FROM wgperformimage
WHERE img_num=#{num}
</select>
```

3. 공연 정보 제공



공연 상세정보

저장 | 목록

유진과 유진

뮤지컬

2022-06-18 ~ 2022-08-28

대학로 자유극장 위치보기

100분

강지혜, 이상아, 홍나현, 이아진, 윤

알아주는 모범생 인생을 살며 조용
학교 2학년에 같은 반으로 만나게
한다. 유치원 때 헤어진 작은유진이
다시 무시당한다. 수학여행에서 우
을 잊고자 스스로를 책망하고 그러던 중, 잊고 있던 기억의 파편이
은 진실이 큰유진이 말했던 '그 일'과 연관 있을 거라 느낀다. 결국

마타하리

뮤지컬

2022-05-28 ~ 2022-08-15

샤롯데씨어터 위치보기

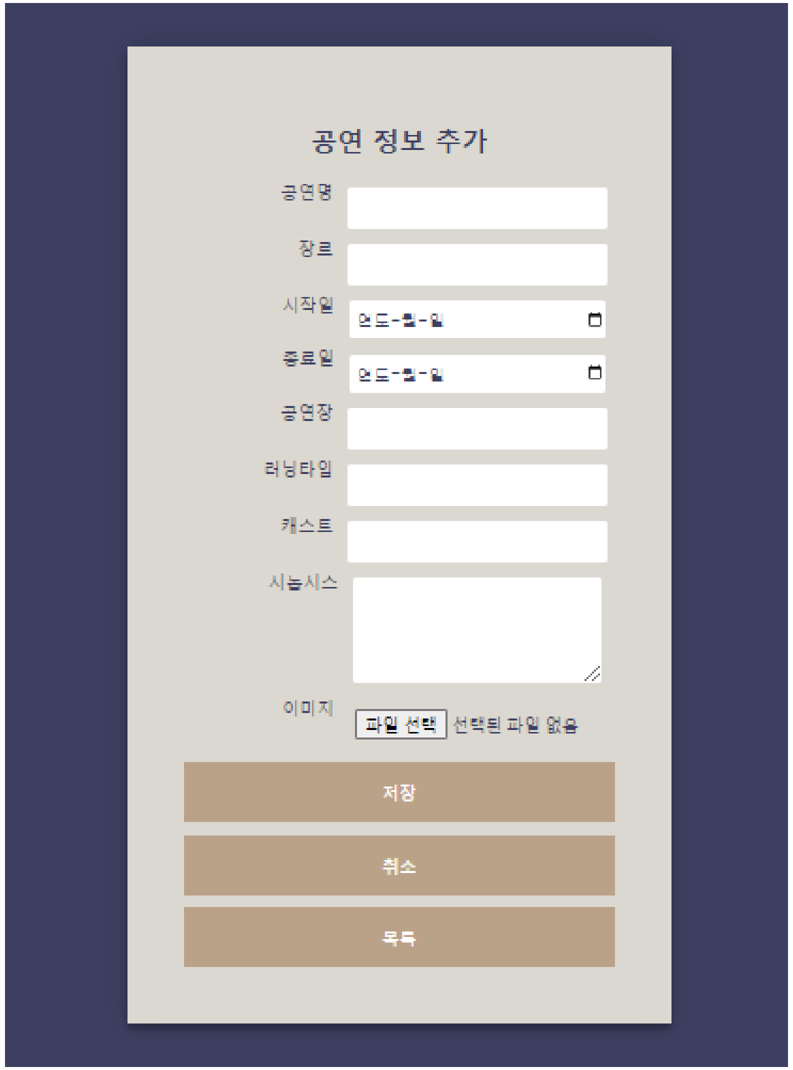
180분

옥주현, 솔라, 김성식, 이홍기, 이창섭, 윤소호, 최민철, 김바울, 한지연, 최나래, 홍경수, 육현욱, 김지혜, 최진

1917년 파리 교외의 어느 들판, 숨을 멎게 할 만큼 아름답고 고혹적인 마타 하리가 서있다. 그녀의 기억을 스쳐가는 사람들과 노래 속에, 9개월 전 물랑루즈의 화려한 무대로 시간은 거슬러 올라간다. 관능적이고 신비로운 매력을 지닌 마타하리는 '사원의 춤'으로 선풍적인 인기를 끌며 유럽 전역의 유명인사로 활동한다. 그러던 어느 날 마타하리를 찾아온 프랑스 정보부의 라두 대령과의 만남이 그녀의 인생을 송두리째 바꿔 놓는다. 유럽의 고위 간부들과 친분을 맺고 있으며, 유명 인사라는 신분 덕분에 전쟁 중에도 국경을 자유롭게 넘나들 수 있는 마타하리는 세계 1차대전에서 불리한 상황에 놓인 프랑스에게 더 없이 유용한 존재였고 라두 대령은 그녀에게 프랑스의 스파이가 되기를 요구한다. 한편 마타하리는 센느 강변에서 조종사 아르망과 우연히 마주치게 되고, 사랑에 빠지고 만다. 전쟁의 기운이 짙어지는 가운데, 마타하리를 스파이로 고용한 라두 대령은 점점 아름다운 마타하리에게 매료되고 라두 대령과 아르망 사이에 묘한 긴장감이 흐르게 된다. 결국 라두 대령은 아르망을 현재 독일의 점령지이자 위험 지역인 비텔(Vittel)로 보낸다. 생사여부가 불투명해진 아르망의 행방을 확인하기 위해, 그녀는 위조된 서류를 가지고 위험을 무릅쓴 채 비텔로 향하는데......

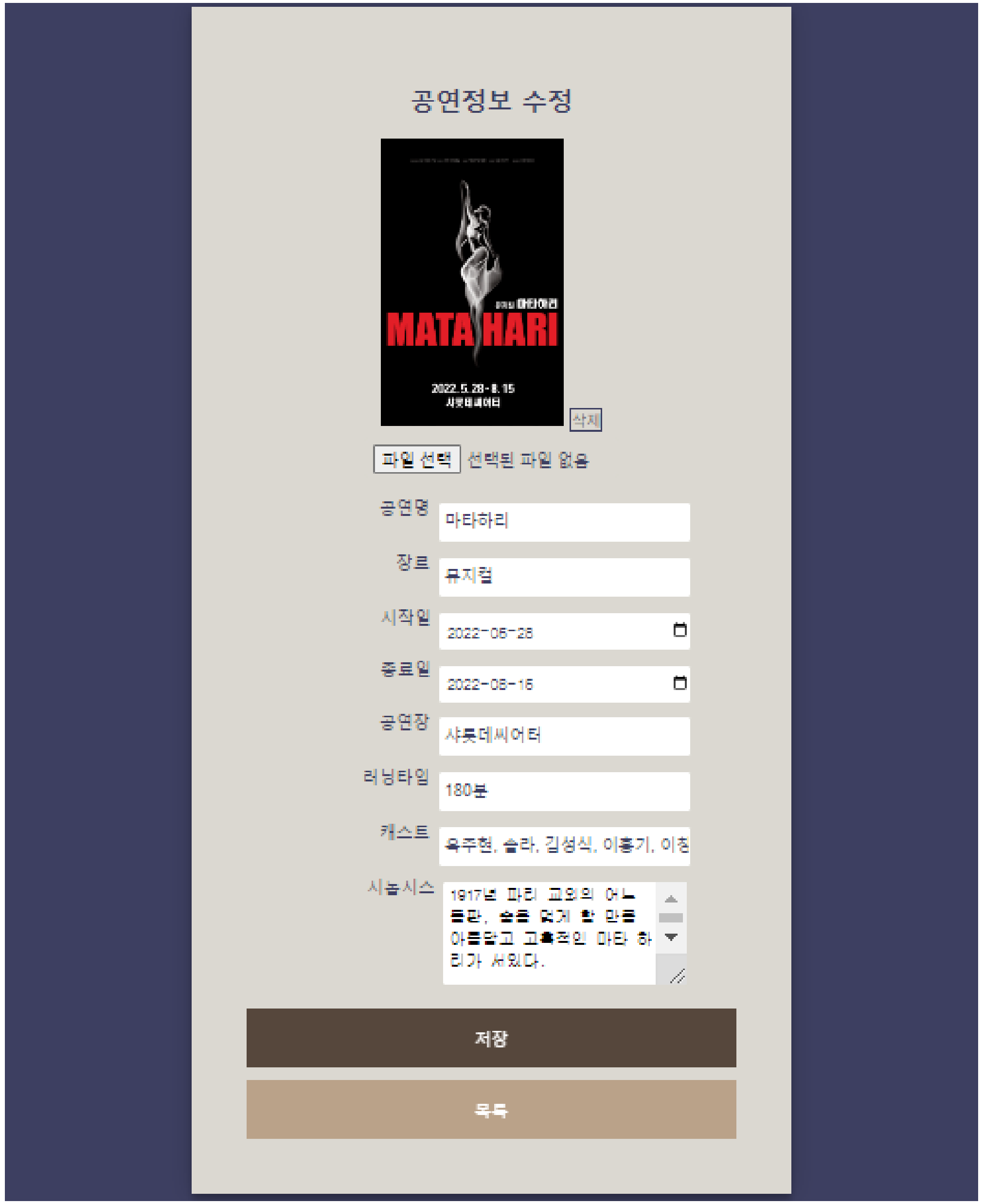
2022 뮤지컬 마타하리 (Musical MATA...

3. 공연 정보 제공



공연 정보 추가

공연명	
장르	
시작일	연도-월-일
종료일	연도-월-일
공연장	
러닝타임	
캐스트	
시놉시스	
이미지	파일 선택 선택된 파일 없음



공연정보 수정

공연명	마타하리
장르	뮤지컬
시작일	2022-05-28
종료일	2022-08-15
공연장	샤롯데씨어터
러닝타임	180분
캐스트	옥주현, 솔라, 김성식, 이홍기, 이창
시놉시스	1917년 파리 교외의 어느 들판, 숨을 멎게 할 만큼 아름답고 고혹적인 마타 하리가 서있다.

3. 공연 정보 제공

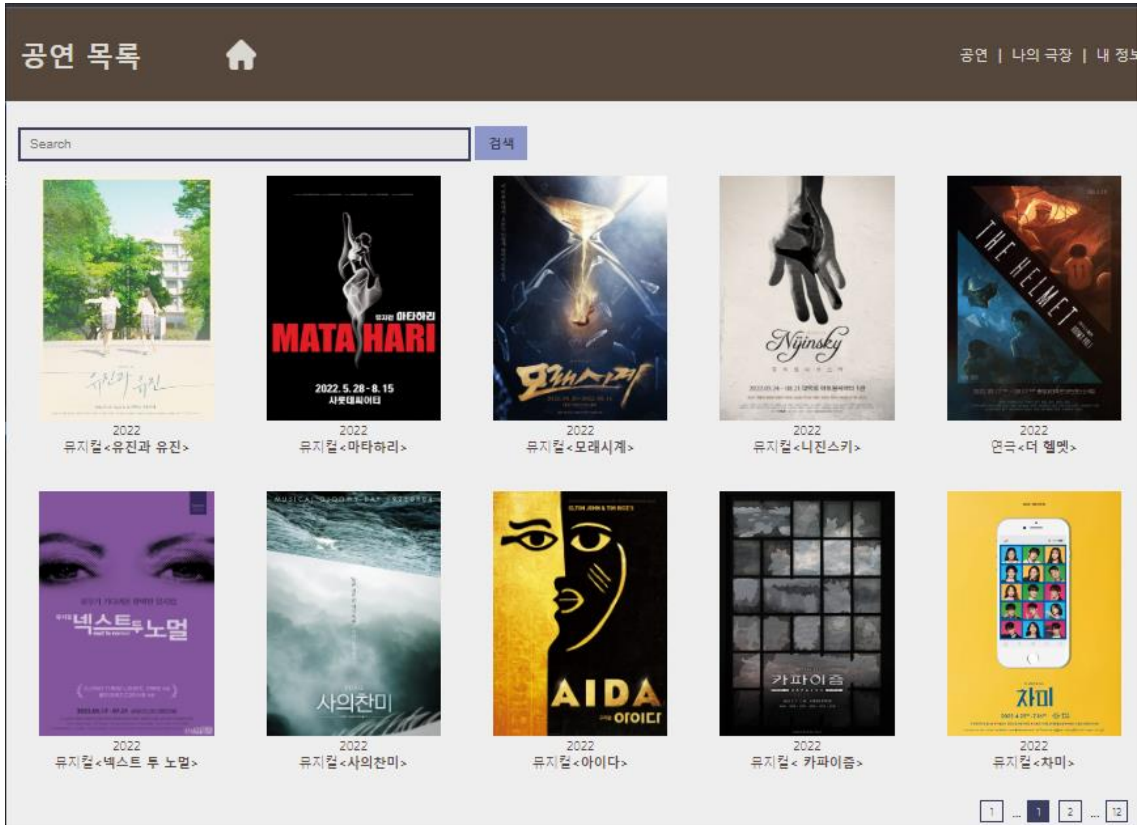
Controller

```
@PostMapping("/file/delete")
@ResponseBody
public Map<String, Boolean> delete_file(@RequestParam int
num){
boolean deleted = svc.delete_file(num, resourceLoader);
Map<String, Boolean> map = new HashMap<>();
map.put("deleted", deleted);
return map;
}

@PostMapping("/deletePerform")
@ResponseBody
public Map<String, Boolean> deletePerform(@RequestParam
int num){
boolean deletedPerform =svc.deletePerform(num,
resourceLoader);
Map<String, Boolean> map = new HashMap<>();
map.put("deletedPerform", deletedPerform);
return map;
}
```

Controller

```
@GetMapping("/detail")
public String detail(@RequestParam int num, Model model) {
WgPerformVO perform = svc.performDetail(num);
model.addAttribute("perform",perform);
return "pj/detailPerform";
}

@PostMapping("/addPerform")
@ResponseBody
public Map<String, Boolean> addPerform(@RequestParam(value="files", required=false) MultipartFile mfiles,
HttpServletRequest request,
WgPerformVO perform){
Map<String, Boolean> map = new HashMap<>();
boolean addedPerform = svc.addPerform(request, perform, mfiles);
map.put("addedPerform", addedPerform);
return map;
}

@PostMapping("/updatePerform")
@ResponseBody
public Map<String, Boolean> updatePerform(@RequestParam(value="files", required=false) MultipartFile mfiles,
HttpServletRequest request, WgPerformVO perform){
boolean updatedPerform = svc.updatePerform(request, perform, mfiles);
Map<String, Boolean> map = new HashMap<>();
map.put("updatedPerform", updatedPerform);
return map;
}
```

4. 공연을 리스트 형태로 제공



2022
뮤지컬<유진과 유진>

2022
뮤지컬<마타하리>

2022
뮤지컬<모래시계>

2022
뮤지컬<니진스키>

2022
연극<더 헬멧>

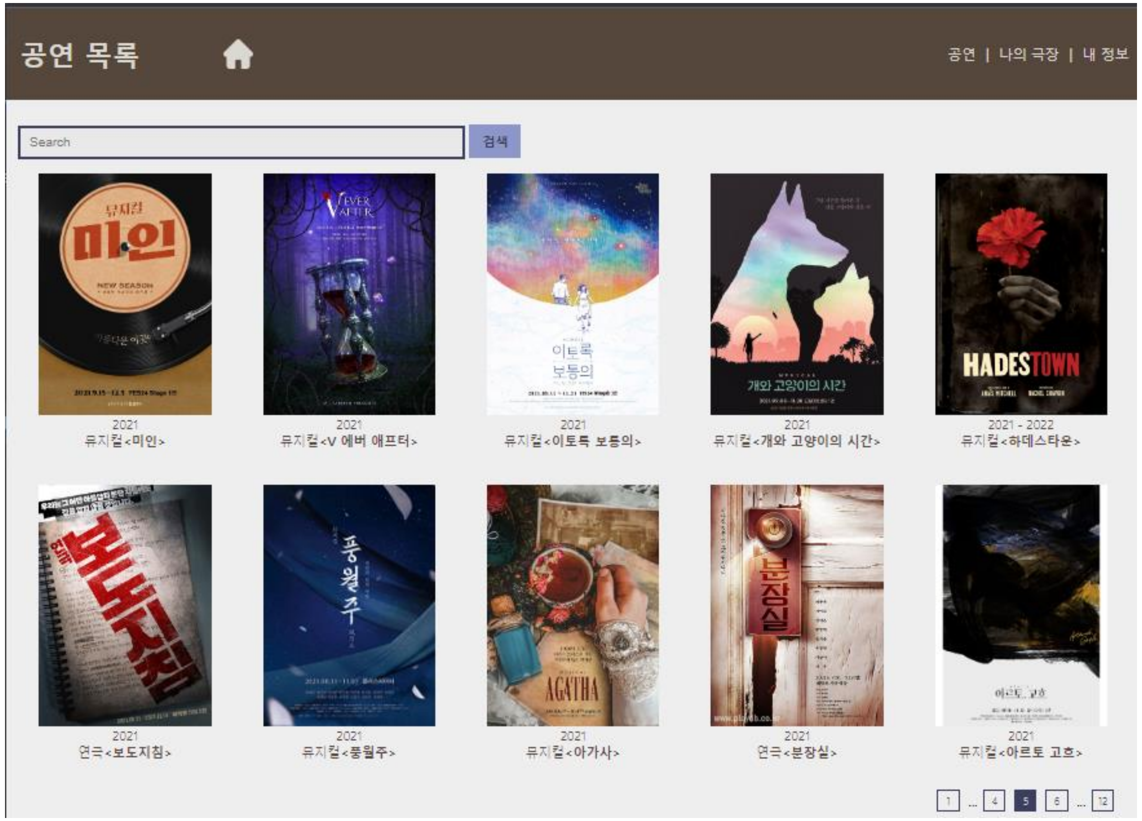
2022
뮤지컬<넥스트 투 노멀>

2022
뮤지컬<사의찬미>

2022
뮤지컬<아이다>

2022
뮤지컬< 카파이즘>

2022
뮤지컬<차미>



2021
뮤지컬<미인>

2021
뮤지컬<V 에버 애프터>

2021
뮤지컬<이토록 보통의>

2021
뮤지컬<개와 고양이의 시간>

2021 - 2022
뮤지컬<하데스타운>

2021
연극<보도지침>

2021
뮤지컬<풍월주>

2021
뮤지컬<아가사>

2021
연극<분장실>

2021
뮤지컬<아르토 고흐>

4. 공연을 리스트 형태로 제공

Controller

```
@GetMapping("/list")
//@ResponseBody
public String getPageSearch(@SessionAttribute(name="uid",
required=false) String uid, @RequestParam(required=false,
defaultValue="1") int pageNum, Model model,
@RequestParam(defaultValue="") String keyword) {
      PageInfo<Map<String, Object>> pageInfo =
svc.getListPageSearch(pageNum, keyword);
      pageInfo.getNavigatepageNums();
model.addAttribute("pageInfo",pageInfo);
      model.addAttribute("list",pageInfo.getList());
      model.addAttribute("keyword",keyword);
      model.addAttribute("uid",uid);
      return "pj/listPageSearch2";
}
```

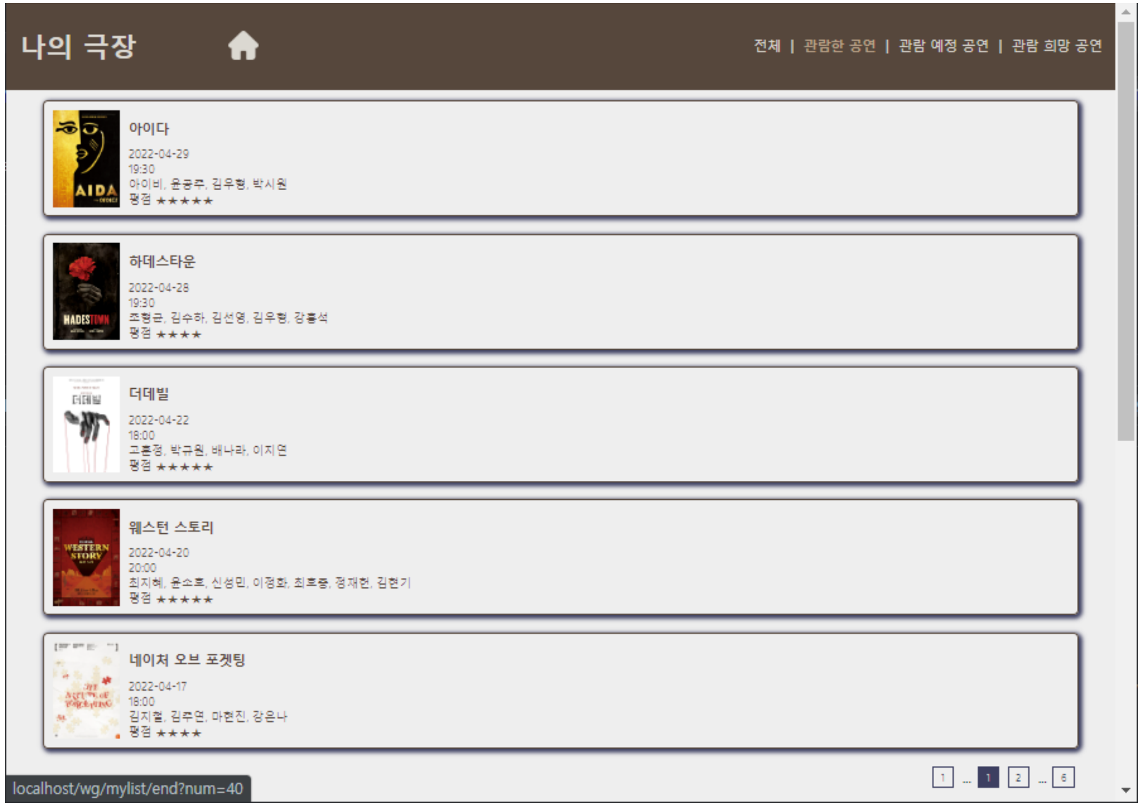
DAO

```
public PageInfo<Map<String, Object>> getListPageSearch(int pageNum,
String keyword) {
      PageHelper.startPage(pageNum, 10);
      PageInfo<Map<String, Object>> pageInfo = new
PageInfo<>(dao.searchPerform(keyword));
      return pageInfo;
}
```

Mapper

```
<select id="performanceList"
resultType="map">
SELECT * FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
ORDER BY p.startdate DESC
</select>

<select id="searchPerform"
resultType="map"
parameterType="String">
SELECT * FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE name LIKE CONCAT('%',#{keyword},'%') OR loc LIKE CONCAT('%',#{keyword},'%') OR cast LIKE
CONCAT('%',#{keyword},'%')
ORDER BY p.startdate DESC
</select>

<select id="getTotal"
resultType="int">
SELECT COUNT(*) FROM wgperformance p LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE name LIKE CONCAT('%',#{keyword},'%') OR loc LIKE CONCAT('%',#{keyword},'%') OR cast LIKE
CONCAT('%',#{keyword},'%')
</select>
```

5. 자신이 본 공연 저장/수정/삭제



아이다
2022-04-29
19:30
아이비, 윤공주, 김우형, 박시원
평점 ★★★★★

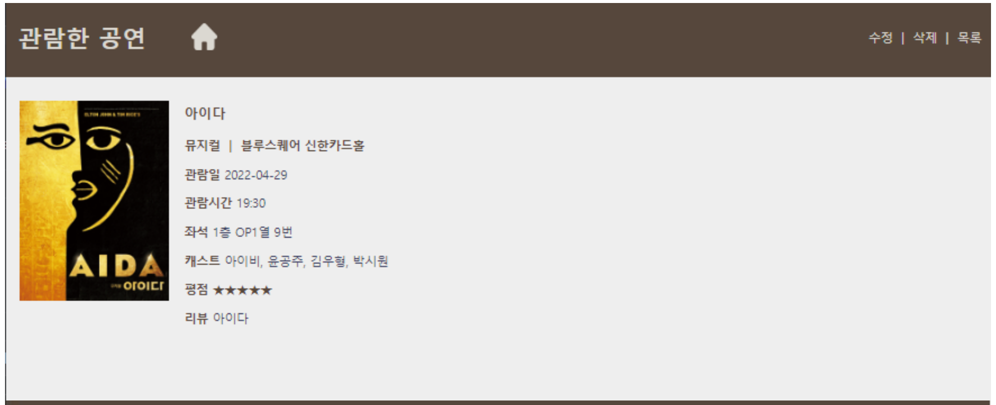
하데스타운
2022-04-28
19:30
조형균, 김수하, 김선영, 김우형, 강홍석
평점 ★★★★

더데빌
2022-04-22
18:00
고훈정, 박규원, 배나라, 이지연
평점 ★★★★★

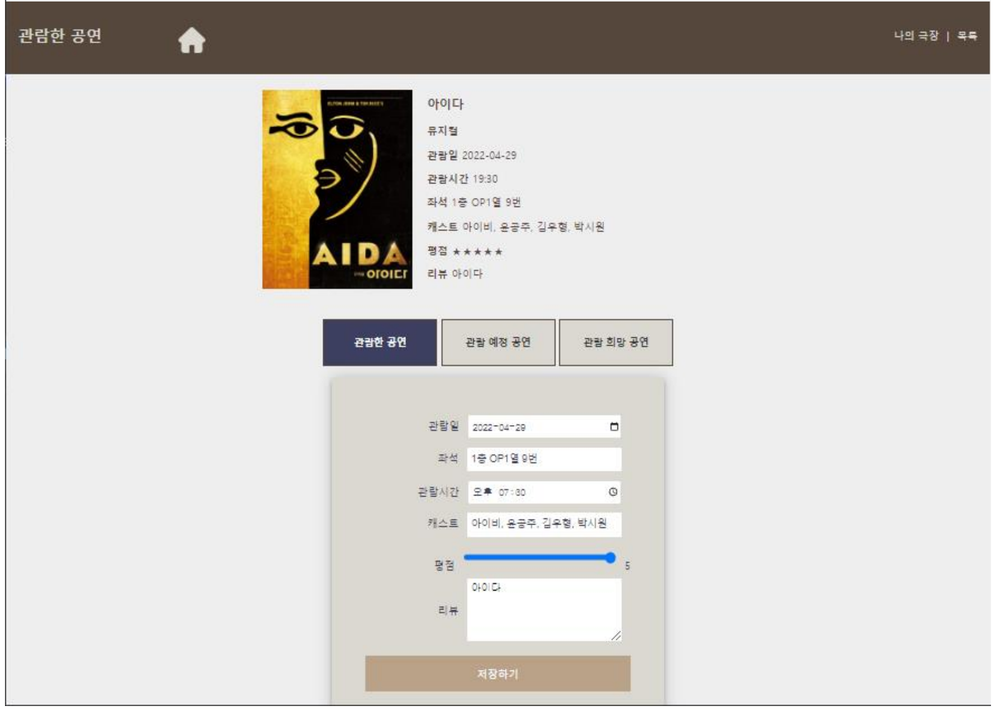
웨스턴 스토리
2022-04-20
20:00
최지혜, 윤소호, 신성민, 이정화, 최호중, 정재헌, 김현기
평점 ★★★★★

네이처 오브 포겟팅
2022-04-17
18:00
김지철, 김주연, 마현진, 강은나
평점 ★★★★

localhost/wg/mylist/end?num=40



아이다
뮤지컬 | 블루스퀘어 신한카드홀
관람일 2022-04-29
관람시간 19:30
좌석 1층 OP1열 9번
캐스트 아이비, 윤공주, 김우형, 박시원
평점 ★★★★★
리뷰 아이다

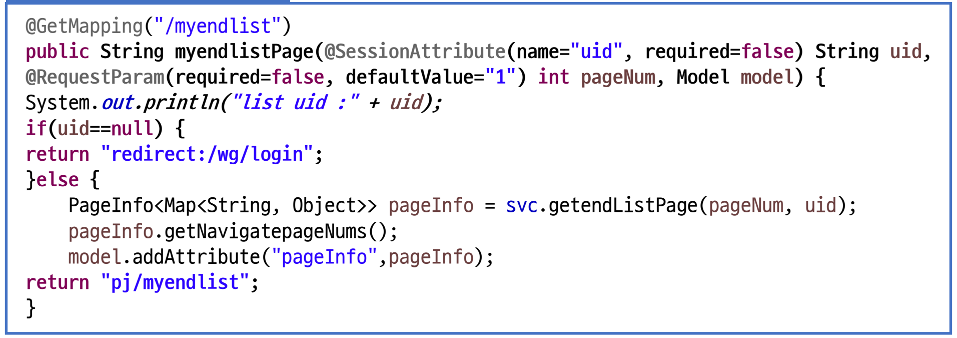


아이다
뮤지컬
관람일 2022-04-29
관람시간 19:30
좌석 1층 OP1열 9번
캐스트 아이비, 윤공주, 김우형, 박시원
평점 ★★★★★
리뷰 아이다

관람일 2022-04-29
좌석 1층 OP1열 9번
관람시간 오후 07:30
캐스트 아이비, 윤공주, 김우형, 박시원
평점 5
리뷰 아이다

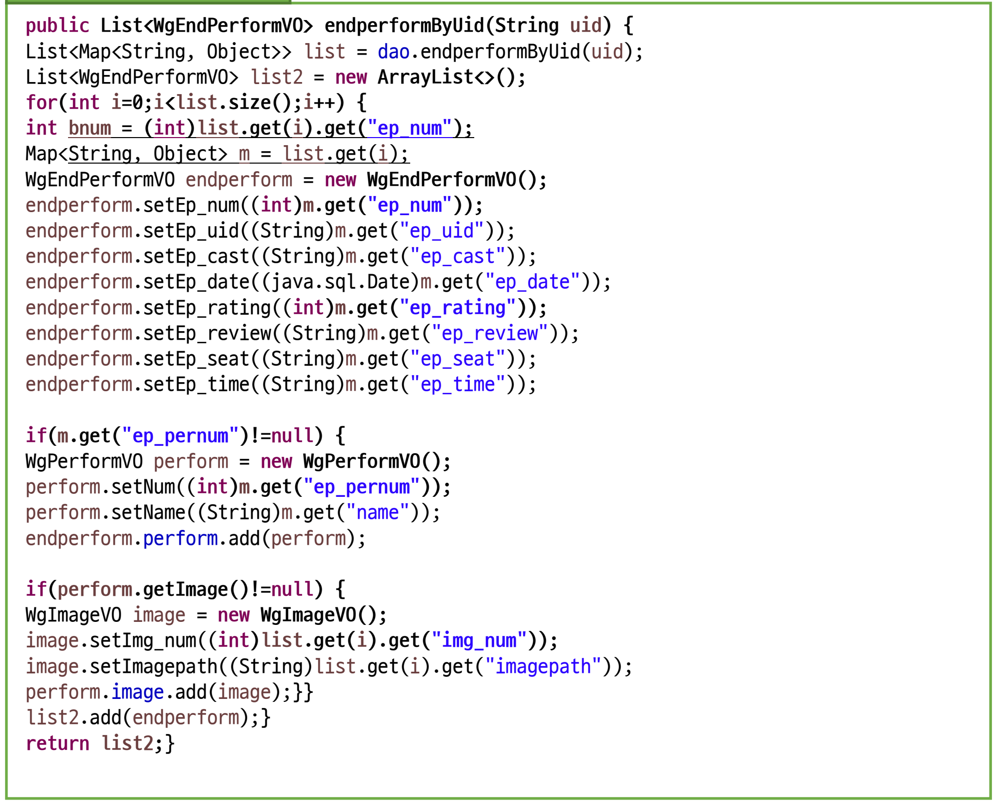
5. 자신이 본 공연 저장/수정/삭제

Controller



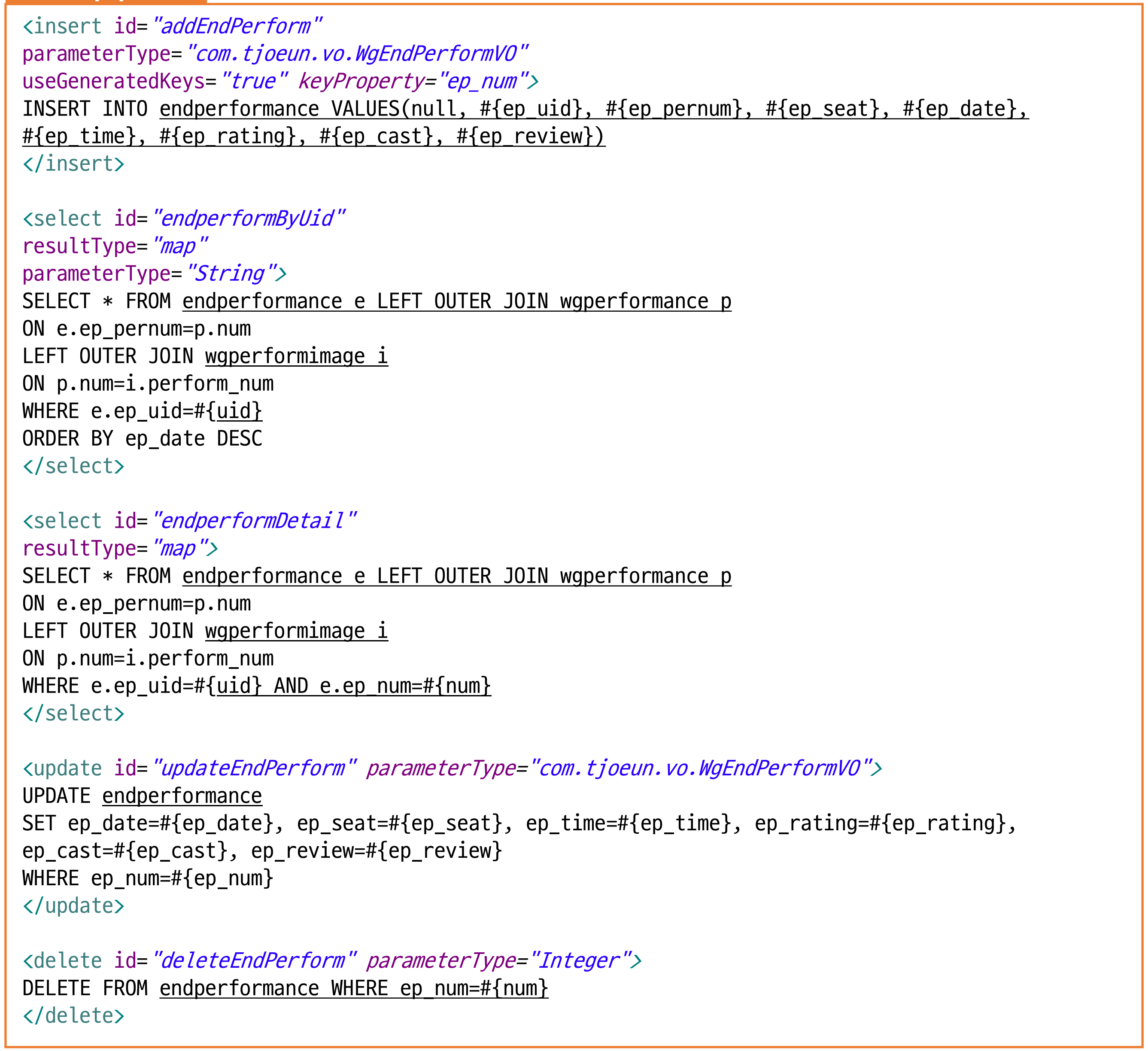
```
@GetMapping("/myendlist")
public String myendlistPage(@SessionAttribute(name="uid", required=false) String uid,
@RequestParam(required=false, defaultValue="1") int pageNum, Model model) {
System.out.println("list uid :" + uid);
if(uid==null) {
return "redirect:/wg/login";
}else {
    PageInfo<Map<String, Object>> pageInfo = svc.getendListPage(pageNum, uid);
    pageInfo.getNavigatepageNums();
    model.addAttribute("pageInfo",pageInfo);
return "pj/myendlist";
}
```

SVC



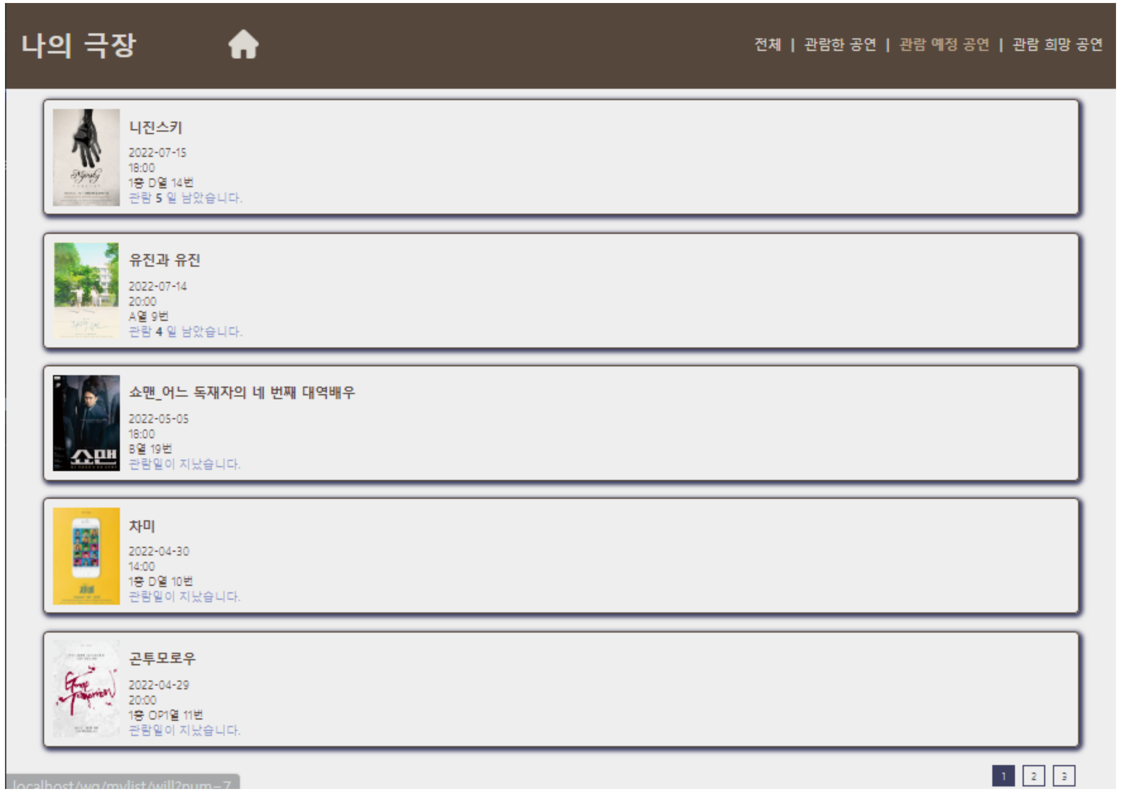
```
public List<WgEndPerformVO> endperformByUid(String uid) {
List<Map<String, Object>> list = dao.endperformByUid(uid);
List<WgEndPerformVO> list2 = new ArrayList<>();
for(int i=0;i<list.size();i++) {
int bnum = (int)list.get(i).get("ep_num");
Map<String, Object> m = list.get(i);
WgEndPerformVO endperform = new WgEndPerformVO();
endperform.setEp_num((int)m.get("ep_num"));
endperform.setEp_uid((String)m.get("ep_uid"));
endperform.setEp_cast((String)m.get("ep_cast"));
endperform.setEp_date((java.sql.Date)m.get("ep_date"));
endperform.setEp_rating((int)m.get("ep_rating"));
endperform.setEp_review((String)m.get("ep_review"));
endperform.setEp_seat((String)m.get("ep_seat"));
endperform.setEp_time((String)m.get("ep_time"));

if(m.get("ep_pernum")!=null) {
WgPerformVO perform = new WgPerformVO();
perform.setNum((int)m.get("ep_pernum"));
perform.setName((String)m.get("name"));
endperform.perform.add(perform);

if(perform.getImage()!=null) {
WgImageVO image = new WgImageVO();
image.setImg_num((int)list.get(i).get("img_num"));
image.setImagepath((String)list.get(i).get("imagepath"));
perform.image.add(image);}}
list2.add(endperform);}
return list2;}
```

Mapper



```
<insert id="addEndPerform"
parameterType="com.tjoeun.vo.WgEndPerformVO"
useGeneratedKeys="true" keyProperty="ep_num">
INSERT INTO endperformance VALUES(null, #{ep_uid}, #{ep_pernum}, #{ep_seat}, #{ep_date},
#{ep_time}, #{ep_rating}, #{ep_cast}, #{ep_review})
</insert>

<select id="endperformByUid"
resultType="map"
parameterType="String">
SELECT * FROM endperformance e LEFT OUTER JOIN wgperformance p
ON e.ep_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE e.ep_uid=#{uid}
ORDER BY ep_date DESC
</select>

<select id="endperformDetail"
resultType="map">
SELECT * FROM endperformance e LEFT OUTER JOIN wgperformance p
ON e.ep_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE e.ep_uid=#{uid} AND e.ep_num=#{num}
</select>

<update id="updateEndPerform" parameterType="com.tjoeun.vo.WgEndPerformVO">
UPDATE endperformance
SET ep_date=#{ep_date}, ep_seat=#{ep_seat}, ep_time=#{ep_time}, ep_rating=#{ep_rating},
ep_cast=#{ep_cast}, ep_review=#{ep_review}
WHERE ep_num=#{ep_num}
</update>

<delete id="deleteEndPerform" parameterType="Integer">
DELETE FROM endperformance WHERE ep_num=#{num}
</delete>
```

6. 자신이 볼 예정인 공연 저장/수정/삭제


나의 극장
전체 | 관람한 공연 | 관람 예정 공연 | 관람 희망 공연

니진스키
2022-07-15
18:00
1층 D열 14번
관람 5 일 남았습니다.

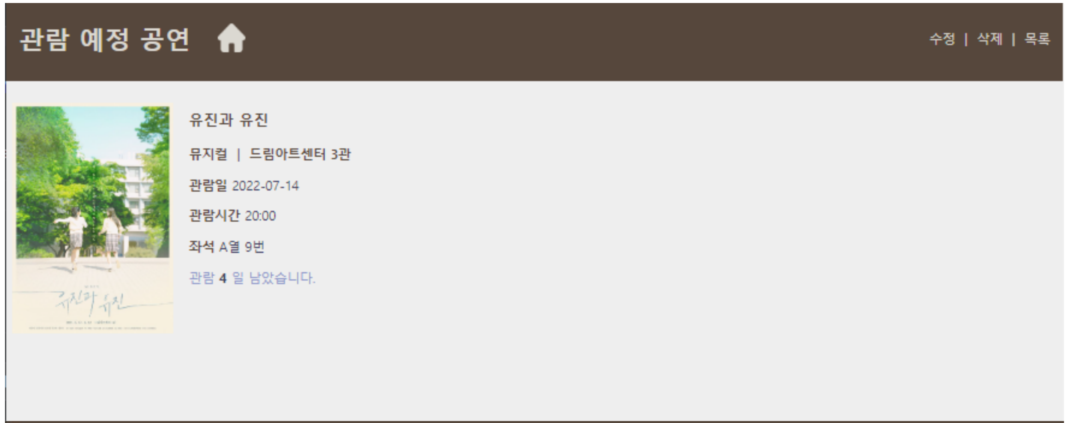
유진과 유진
2022-07-14
20:00
A열 9번
관람 4 일 남았습니다.

쇼맨_어느 독재자의 네 번째 대역배우
2022-05-05
18:00
B열 19번
관람일이 지났습니다.

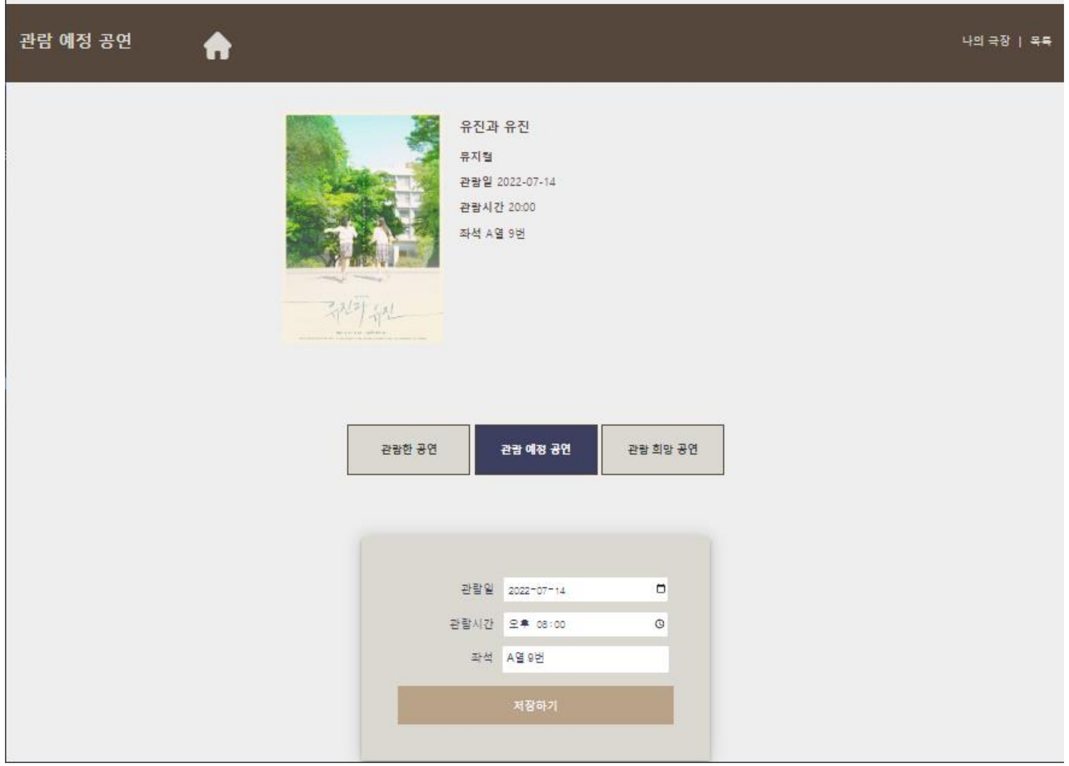
차미
2022-04-30
14:00
1층 D열 10번
관람일이 지났습니다.

곤투모로우
2022-04-29
20:00
1층 OP1열 11번
관람일이 지났습니다.

1 2 3
localhost/wa/mylist/will?num=7



유진과 유진
뮤지컬 | 드림아트센터 3관
관람일 2022-07-14
관람시간 20:00
좌석 A열 9번
관람 4 일 남았습니다.



유진과 유진
뮤지컬
관람일 2022-07-14
관람시간 20:00
좌석 A열 9번

관람일 2022-07-14
관람시간 오후 08:00
좌석 A열 9번

6. 자신이 볼 예정인 공연 저장/수정/삭제

Controller

```
@GetMapping("/mywilllist")
public String mywilllist(@SessionAttribute(name="uid", required=false) String uid,
@RequestParam(required=false, defaultValue="1") int pageNum, Model model) {
System.out.println("list uid :" + uid);
if(uid==null) {
return "redirect:/wg/login";
}else {
//Map<WgWillPerformVO, Integer> Ddays = svc.Ddays(uid);
PageInfo<Map<String, Object>> pageInfo = svc.getwillListPage(pageNum, uid);
    pageInfo.getNavigatepageNums();
    model.addAttribute("pageInfo",pageInfo);
//    model.addAttribute("Ddays",Ddays);
return "pj/mywilllist";
}
}
```

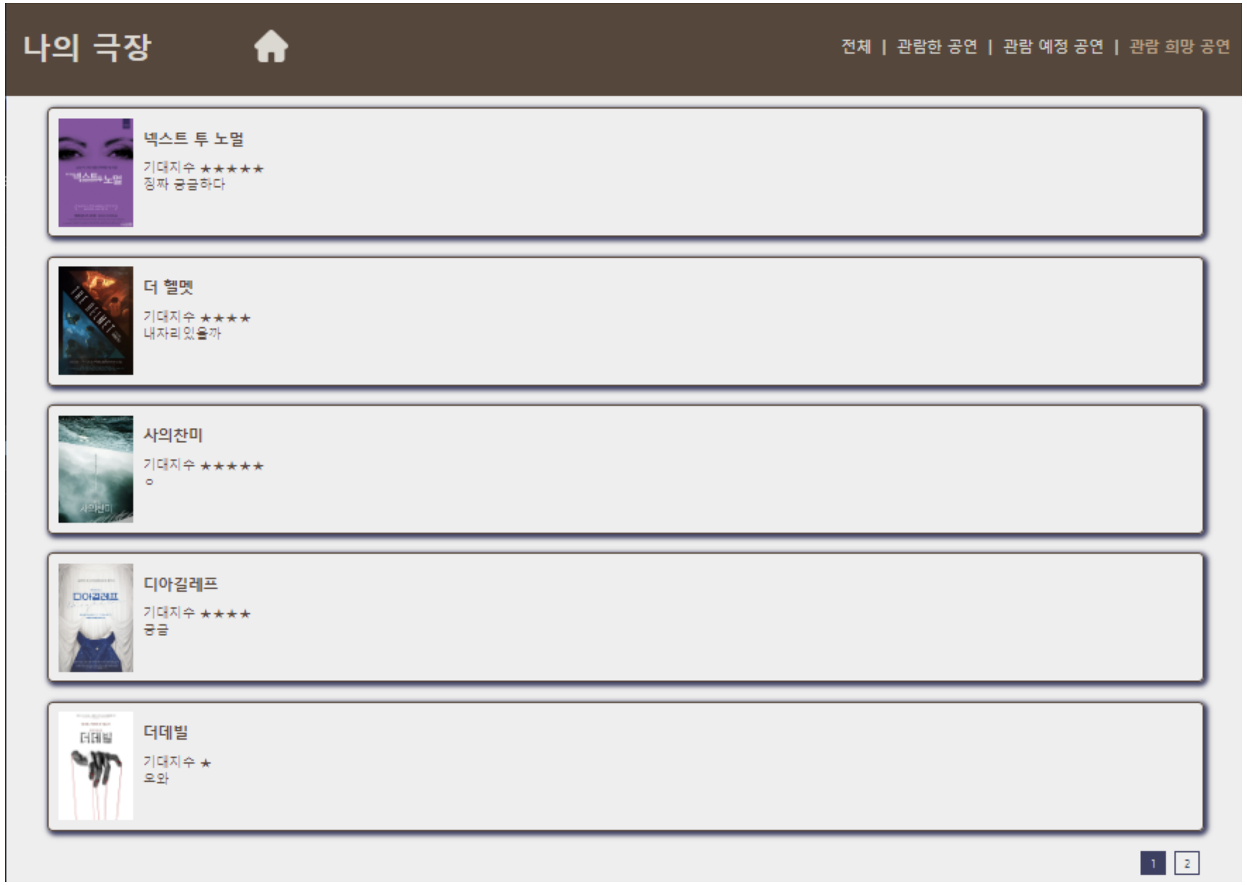
SVC

```
public WgWillPerformVO willperformDetail(String uid, int num) {
List<Map<String, Object>> list = dao.willperformDetail(uid, num);
WgWillPerformVO willperform = new WgWillPerformVO();
int size = list.size();
for(int i=0;i<size;i++) {
Map<String, Object> map = list.get(i);
if(i==0) { // 루프 초기에 한번만 글 정보를 추출
willperform.setWp_uid((String)map.get("wp_uid"));
willperform.setWp_num((int)map.get("wp_num"));
willperform.setWp_date((java.sql.Date)map.get("wp_date"));
willperform.setWp_seat((String)map.get("wp_seat"));
willperform.setWp_time((String)map.get("wp_time"));
willperform.setWp_pernum((int)map.get("wp_pernum"));

}
Object obj = map.get("wp_pernum");
if(obj!=null) {
WgPerformVO perform = new WgPerformVO();
perform.setName((String)map.get("name"));
perform.setNum((int)map.get("wp_pernum"));
perform.setGenre((String)map.get("genre"));
perform.setLoc((String)map.get("loc"));
willperform.perform.add(perform);

Object obj2 = map.get("imagepath");
if(obj2 !=null) {
WgImageVO image = new WgImageVO();
image.setImagepath((String)map.get("imagepath"));
image.setImg_num((int)map.get("img_num"));
perform.image.add(image);
}
}
}
return willperform;
}
```

Mapper

```
<select id="willperformByUid"
resultType="map"
parameterType="String">
SELECT * FROM willperformance w LEFT OUTER JOIN wgperformance p
ON w.wp_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE w.wp_uid=#{uid}
ORDER BY wp_date DESC
</select>

<select id="willperformDetail"
resultType="map">
SELECT * FROM willperformance w LEFT OUTER JOIN wgperformance p
ON w.wp_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE w.wp_uid=#{uid} AND w.wp_num=#{num}
</select>

<update id="updateWillPerform" parameterType="com.tjoeun.vo.WgWillPerformVO">
UPDATE willperformance
SET wp_date=#{wp_date}, wp_seat=#{wp_seat}, wp_time=#{wp_time}
WHERE wp_num=#{wp_num}
</update>

<delete id="deleteWillPerform" parameterType="Integer">
DELETE FROM willperformance WHERE wp_num=#{num}
</delete>
```

7. 자신이 보고 싶은 공연 저장/수정/삭제



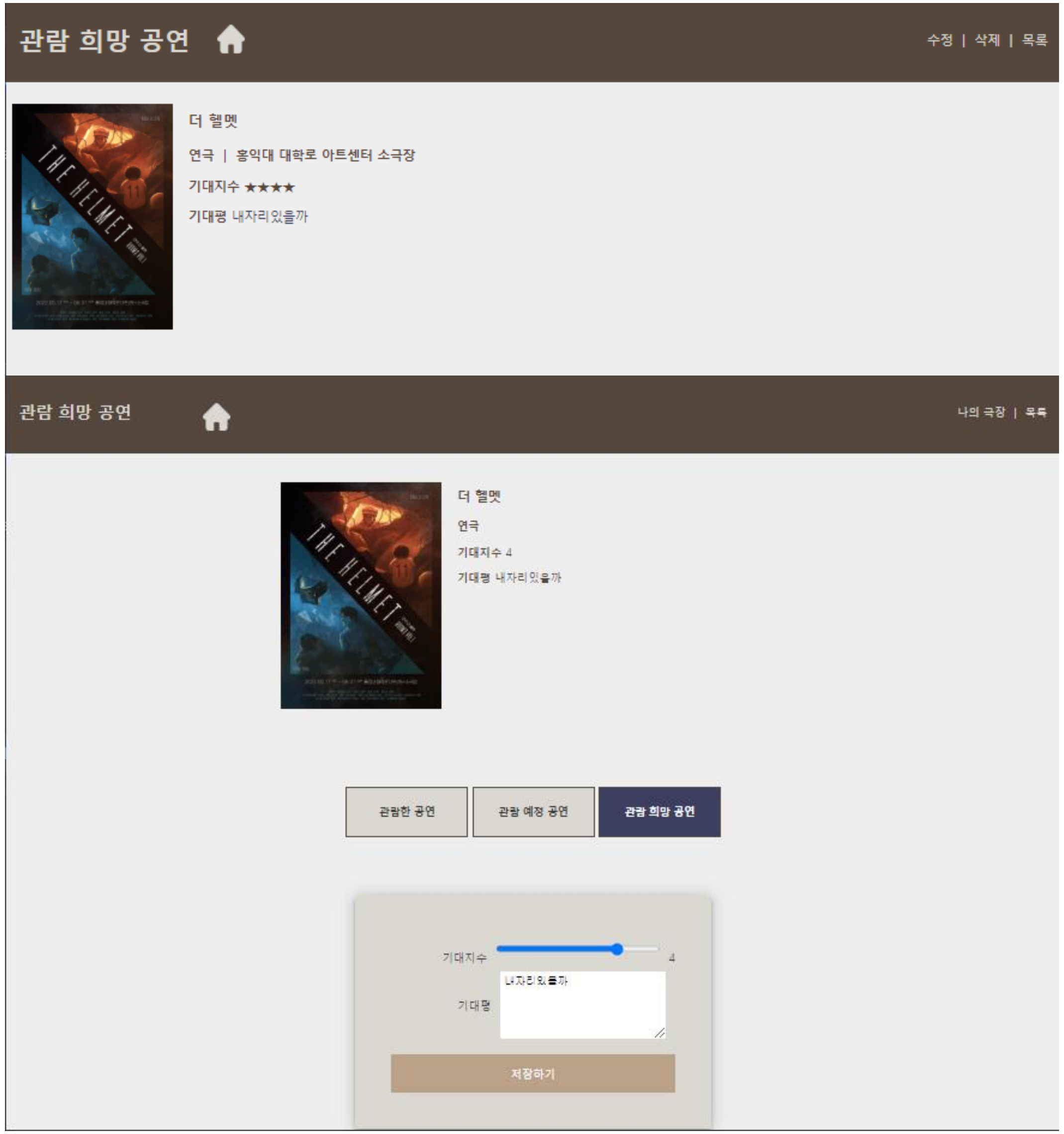
넥스트 투 노멀
기대지수 ★★★★★
징짜 궁금하다

더 헬멧
기대지수 ★★★★
내자리 있을까

사의찬미
기대지수 ★★★★★
ㅇ

디아길레프
기대지수 ★★★★
궁금

더데빌
기대지수 ★
오와



더 헬멧
연극 | 홍익대 대학로 아트센터 소극장
기대지수 ★★★★
기대평 내자리있을까

더 헬멧
연극
기대지수 4
기대평 내자리있을까

기대지수 4
기대평 내자리있을까

7. 자신이 보고 싶은 공연 저장/수정/삭제

Controller

```
@GetMapping("/myhopelist")
public String myhopelist(@SessionAttribute(name="uid", required=false) String uid,
@RequestParam(required=false, defaultValue="1") int pageNum, Model model) {
System.out.println("list uid :" + uid);
if(uid==null) {
return "redirect:/wg/login";
}else {
    PageInfo<Map<String, Object>> pageInfo = svc.gethopeListPage(pageNum, uid);
    pageInfo.getNavigatepageNums();
    model.addAttribute("pageInfo",pageInfo);
return "pj/myhopelist";
}
}
```

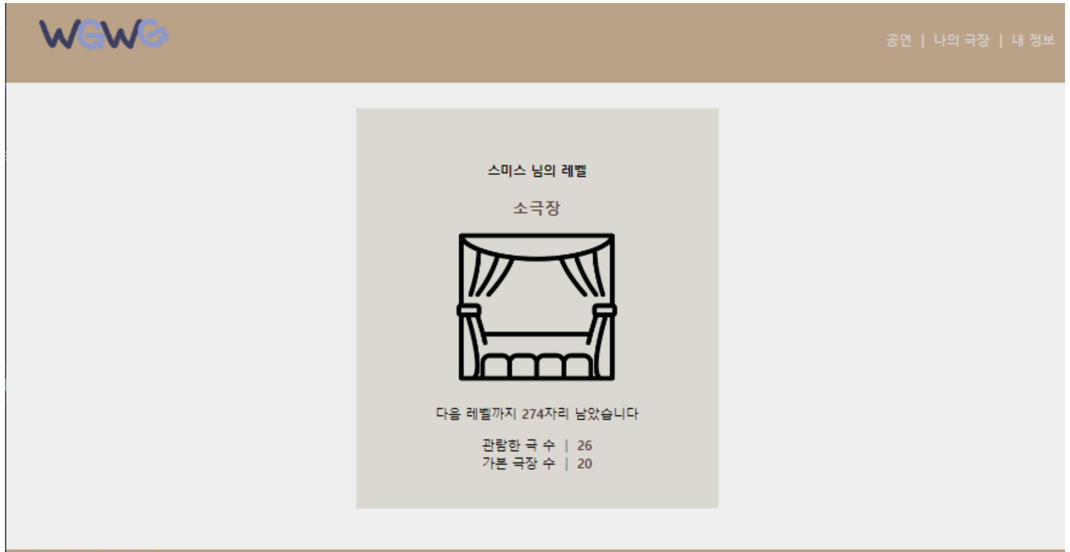
SVC

```
public WgHopePerformVO hopeperformDetail(String uid, int num) {
List<Map<String, Object>> list = dao.hopeperformDetail(uid, num);
WgHopePerformVO hopeperform = new WgHopePerformVO();
int size = list.size();
for(int i=0;i<size;i++) {
Map<String, Object> map = list.get(i);
if(i==0) { // 루프 초기에 한번만 글 정보를 추출
hopeperform.setHp_uid((String)map.get("hp_uid"));
hopeperform.setHp_num((int)map.get("hp_num"));
hopeperform.setHp_rating((int)map.get("hp_rating"));
hopeperform.setHp_review((String)map.get("hp_review"));
hopeperform.setHp_pernum((int)map.get("hp_pernum"));
}
Object obj = map.get("hp_pernum");
if(obj!=null) {
WgPerformVO perform = new WgPerformVO();
perform.setName((String)map.get("name"));
perform.setNum((int)map.get("hp_pernum"));
perform.setGenre((String)map.get("genre"));
perform.setLoc((String)map.get("loc"));
hopeperform.perform.add(perform);

Object obj2 = map.get("imagepath");
if(obj2 !=null) {
WgImageVO image = new WgImageVO();
image.setImagepath((String)map.get("imagepath"));
image.setImg_num((int)map.get("img_num"));
perform.image.add(image);}}}
return hopeperform;}
```

Mapper

```
<insert id="addHopePerform"
parameterType="com.tjoeun.vo.WgHopePerformVO"
useGeneratedKeys="true" keyProperty="hp_num">
INSERT INTO hopeperformance VALUES(null, #{hp_uid}, #{hp_pernum}, #{hp_rating}, #{hp_review})
</insert>

<select id="hopeperformByUid"
resultType="map"
parameterType="String">
SELECT * FROM hopeperformance h LEFT OUTER JOIN wgperformance p
ON h.hp_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE h.hp_uid=#{uid}
</select>

<select id="hopeperformDetail"
resultType="map">
SELECT * FROM hopeperformance h LEFT OUTER JOIN wgperformance p
ON h.hp_pernum=p.num
LEFT OUTER JOIN wgperformimage i
ON p.num=i.perform_num
WHERE h.hp_uid=#{uid} AND h.hp_num=#{num}
</select>

<update id="updateHopePerform" parameterType="com.tjoeun.vo.WgHopePerformVO">
UPDATE hopeperformance
SET hp_rating=#{hp_rating}, hp_review=#{hp_review}
WHERE hp_num=#{hp_num}
</update>

<delete id="deleteHopePerform" parameterType="Integer">
DELETE FROM hopeperformance WHERE hp_num=#{num}
</delete>
```

8. 공연 정보를 누적하여 차트로 제공

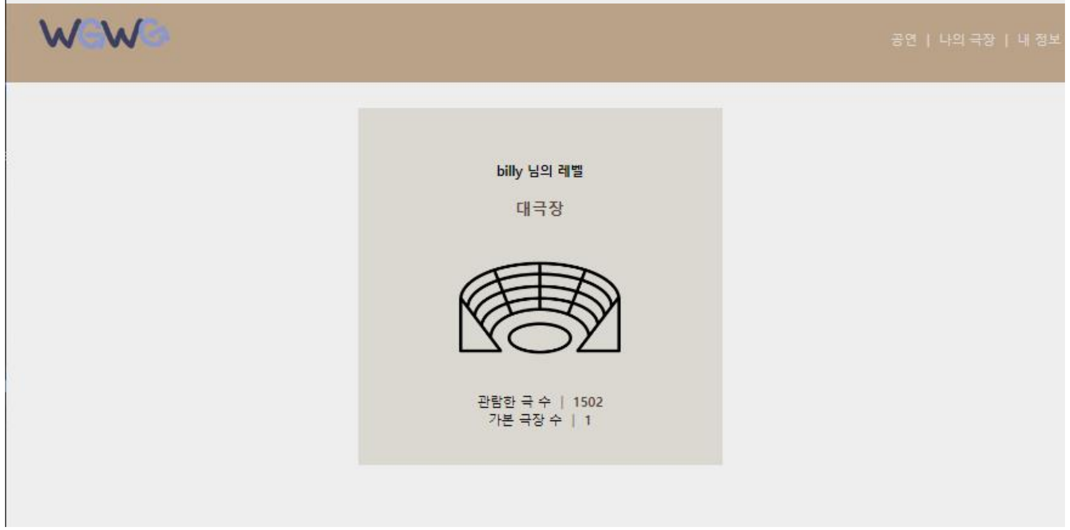


스미스 님의 레벨

소극장

다음 레벨까지 274자리 남았습니다

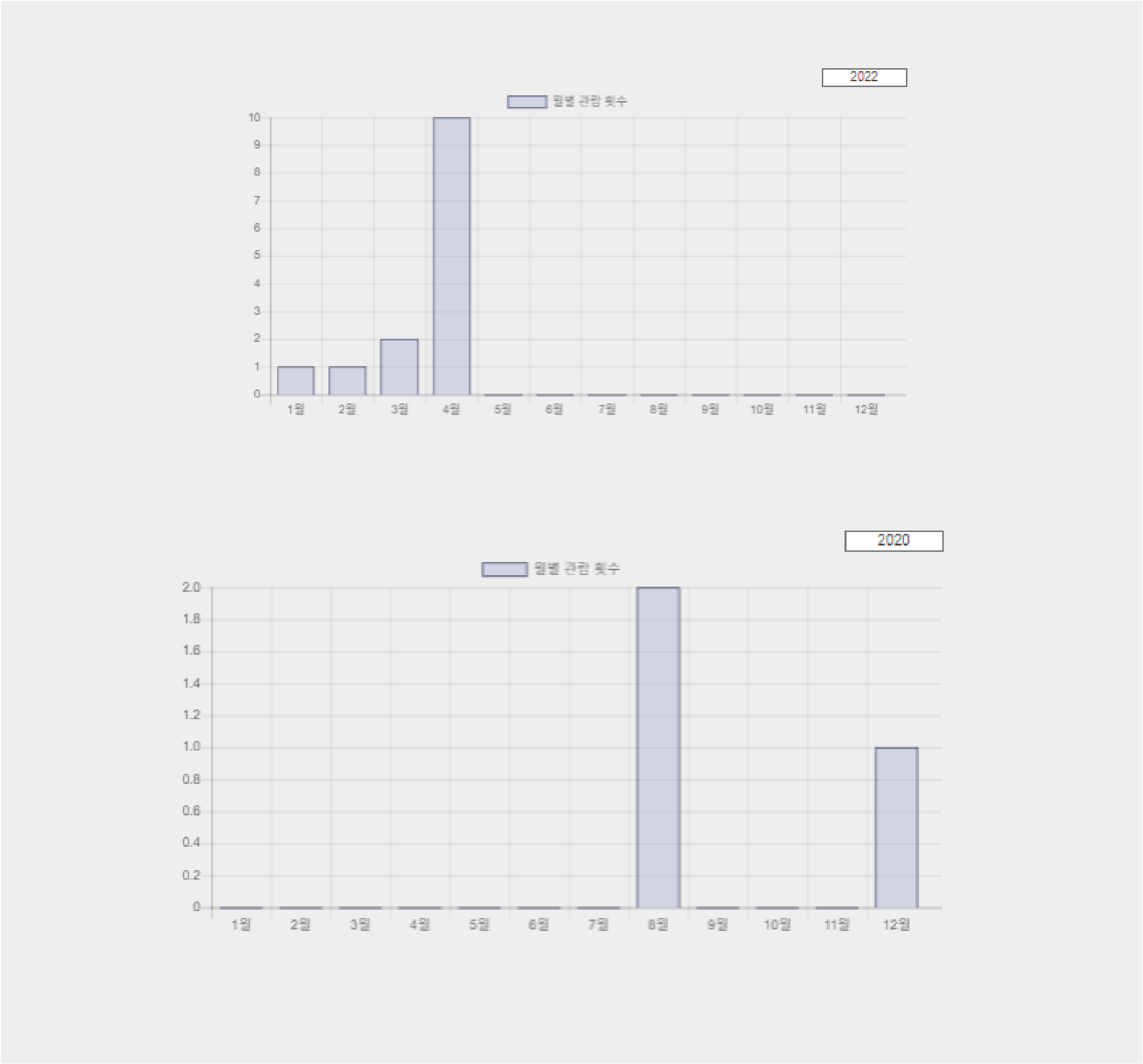
관람한 극 수 | 26
가본 극장 수 | 20



billy 님의 레벨

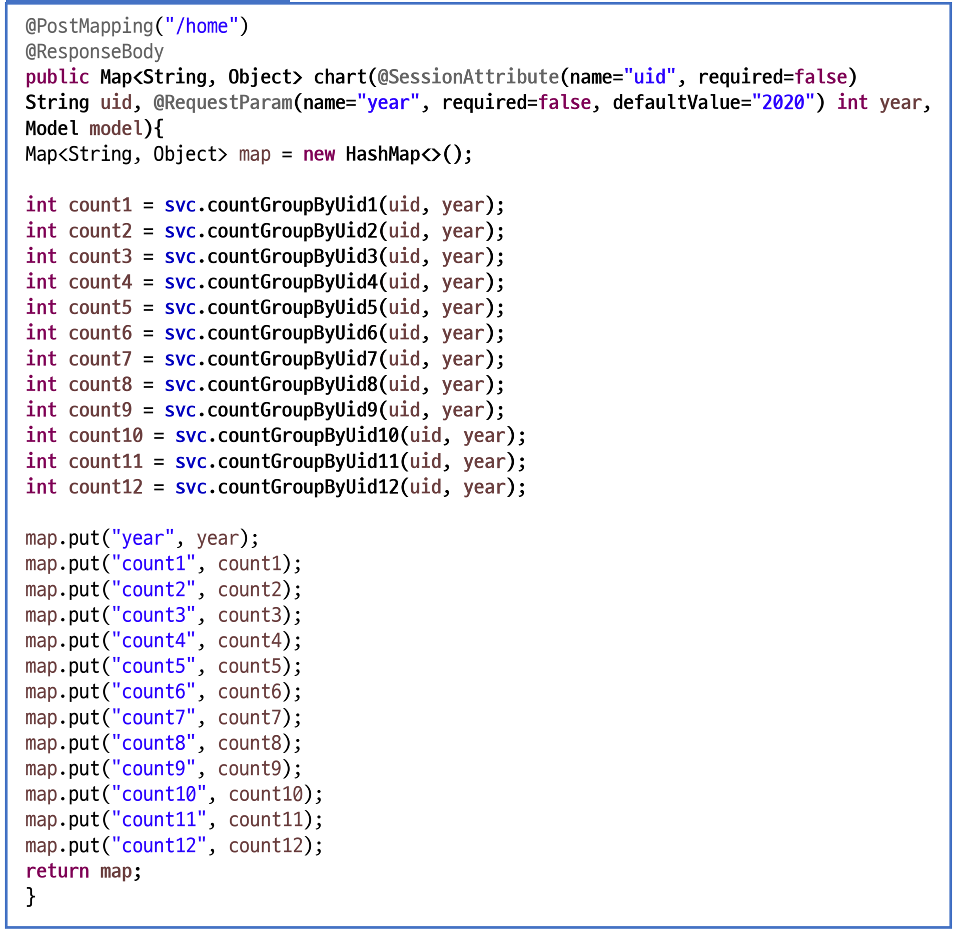
대극장

관람한 극 수 | 1502
가본 극장 수 | 1



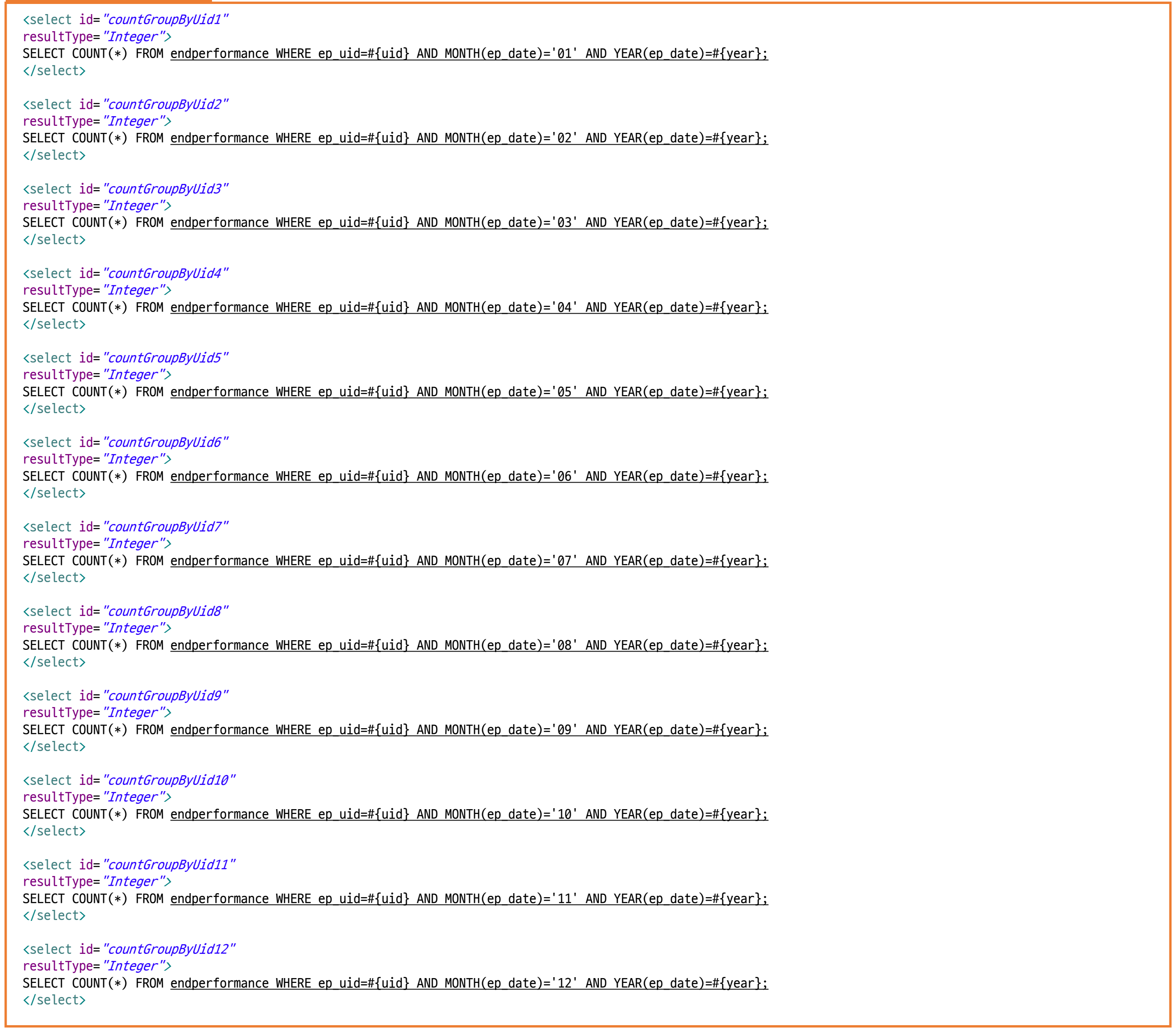
8. 공연 정보를 누적하여 차트로 제공

Controller



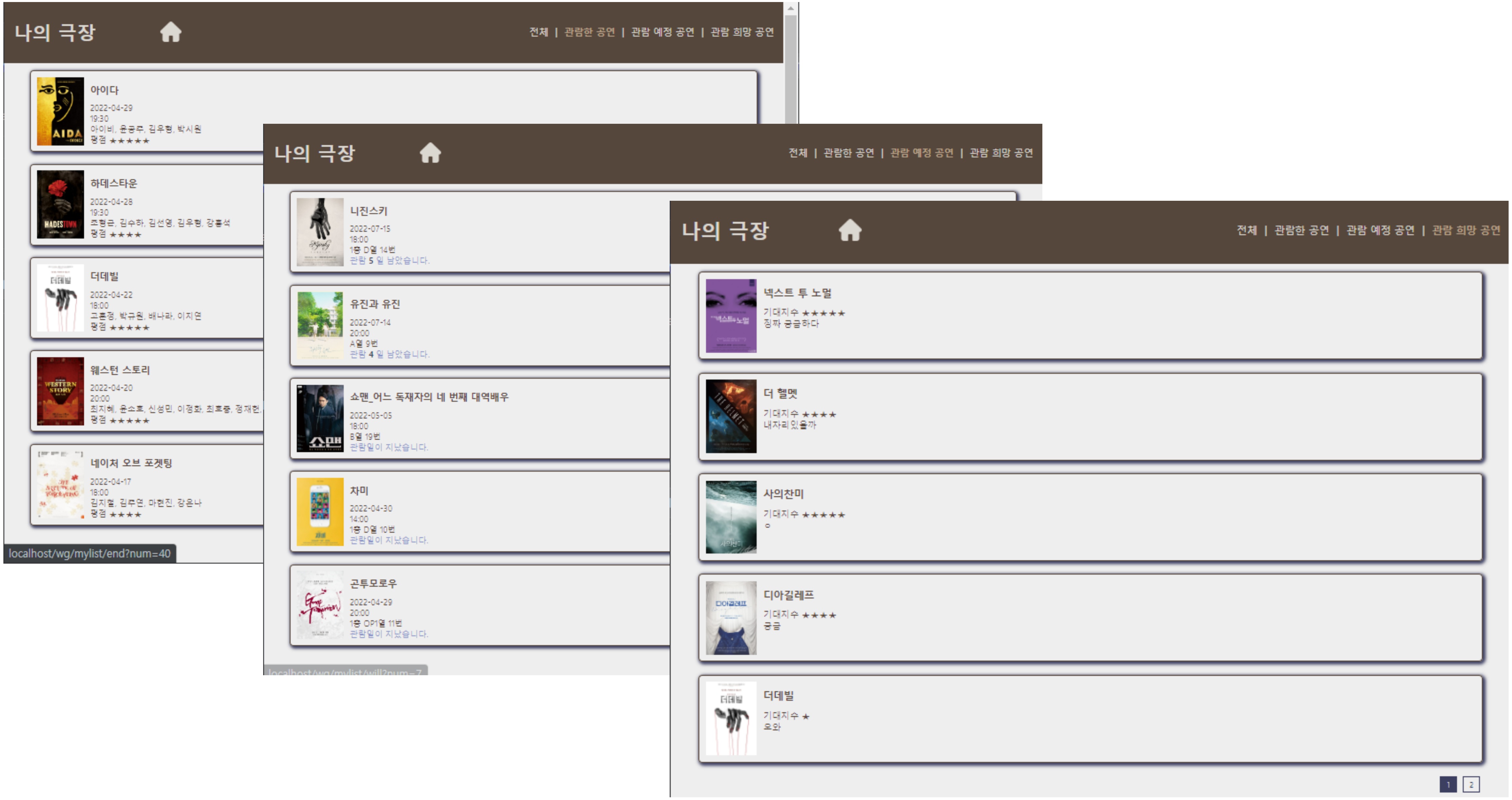
```
@PostMapping("/home")
@ResponseBody
public Map<String, Object> chart(@SessionAttribute(name="uid", required=false)
String uid, @RequestParam(name="year", required=false, defaultValue="2020") int year,
Model model){
Map<String, Object> map = new HashMap<>();

int count1 = svc.countGroupByUid1(uid, year);
int count2 = svc.countGroupByUid2(uid, year);
int count3 = svc.countGroupByUid3(uid, year);
int count4 = svc.countGroupByUid4(uid, year);
int count5 = svc.countGroupByUid5(uid, year);
int count6 = svc.countGroupByUid6(uid, year);
int count7 = svc.countGroupByUid7(uid, year);
int count8 = svc.countGroupByUid8(uid, year);
int count9 = svc.countGroupByUid9(uid, year);
int count10 = svc.countGroupByUid10(uid, year);
int count11 = svc.countGroupByUid11(uid, year);
int count12 = svc.countGroupByUid12(uid, year);

map.put("year", year);
map.put("count1", count1);
map.put("count2", count2);
map.put("count3", count3);
map.put("count4", count4);
map.put("count5", count5);
map.put("count6", count6);
map.put("count7", count7);
map.put("count8", count8);
map.put("count9", count9);
map.put("count10", count10);
map.put("count11", count11);
map.put("count12", count12);
return map;
}
```

Mapper



```
<select id="countGroupByUid1"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='01' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid2"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='02' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid3"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='03' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid4"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='04' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid5"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='05' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid6"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='06' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid7"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='07' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid8"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='08' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid9"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='09' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid10"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='10' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid11"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='11' AND YEAR(ep_date)=#{year};
</select>

<select id="countGroupByUid12"
resultType="Integer">
SELECT COUNT(*) FROM endperformance WHERE ep_uid=#{uid} AND MONTH(ep_date)='12' AND YEAR(ep_date)=#{year};
</select>
```

9. 자신이 본 공연을 리스트 형태로 제공


나의 극장
전체 | 관람한 공연 | 관람 예정 공연 | 관람 희망 공연

아이다
2022-04-29
19:30
아이비, 윤공주, 김우형, 박시원
평점 ★★★★★

하데스타운
2022-04-28
19:30
조형균, 김수하, 김선영, 김우형, 강홍석
평점 ★★★★

더데빌
2022-04-22
18:00
고훈정, 박규원, 배나라, 이지연
평점 ★★★★★

웨스턴 스토리
2022-04-20
20:00
최지혜, 윤소호, 신성민, 이정화, 최호중, 정재헌,
평점 ★★★★★

네이처 오브 포겟팅
2022-04-17
18:00
김지철, 김주연, 마현진, 강은나
평점 ★★★★

localhost/wg/mylist/end?num=40

나의 극장
전체 | 관람한 공연 | 관람 예정 공연 | 관람 희망 공연

니진스키
2022-07-15
18:00
1층 D열 14번
관람 5 일 남았습니다.

유진과 유진
2022-07-14
20:00
A열 9번
관람 4 일 남았습니다.

쇼맨_어느 독재자의 네 번째 대역배우
2022-05-05
18:00
B열 19번
관람일이 지났습니다.

차미
2022-04-30
14:00
1층 D열 10번
관람일이 지났습니다.

곤투모로우
2022-04-29
20:00
1층 OP1열 11번
관람일이 지났습니다.

localhost/wg/mylist/will?num=7

넥스트 투 노멀
기대지수 ★★★★★
징짜 궁금하다

더 헬멧
기대지수 ★★★★
내자리 있을까

사의찬미
기대지수 ★★★★★
ㅇ

디아길레프
기대지수 ★★★★
궁금

더데빌
기대지수 ★
오와

9. 자신이 본 공연을 리스트 형태로 제공

SVC

```
Public List<WgHopePerformVO> hopeperformByUid(String uid) {
List<Map<String, Object>> list = dao.hopeperformByUid(uid);
List<WgHopePerformVO> list2 = new ArrayList<>();
for(int i=0;i<list.size();i++) {
int bnum = (int)list.get(i).get("hp_num");
Map<String, Object> m = list.get(i);
WgHopePerformVO hopeperform = new WgHopePerformVO();
hopeperform.setHp_num((int)m.get("hp_num"));
hopeperform.setHp_uid((String)m.get("hp_uid"));
hopeperform.setHp_rating((int)m.get("hp_rating"));
hopeperform.setHp_review((String)m.get("hp_review"));


if(m.get("hp_pernum")!=null) {

WgPerformVO perform = new WgPerformVO();
perform.setNum((int)m.get("hp_pernum"));
perform.setName((String)m.get("name"));
perform.setGenre((String)m.get("genre"));
hopeperform.perform.add(perform);

if(perform.getImage()!=null) {

WgImageVO image = new WgImageVO();
image.setImg_num((int)list.get(i).get("img_num"));
image.setImagepath((String)list.get(i).get("imagepath"));
System.out.println("이미지패스
"+(String)list.get(i).get("imagepath"));
perform.image.add(image);
}
}
list2.add(hopeperform);
}
return list2;
}
```

프로젝트 시연 영상

https://youtu.be/J3Lp54zamIE

감사합니다

송나은