

{노력을 멈추지 않는  
개발자가 되겠습니다.}

# 프로젝트 소개

진행한 프로젝트에서  
제가 맡은 부분의 일부를 설명하였습니다.



## 연습실 예약 플랫폼 사이트

### Oracle Scheduler

예약 상태 변경

2- 3

### 체크박스 데이터 처리

사용자가 체크박스에 입력한 데이터 DB저장 방식

4

### 이미지 데이터 처리

이미지 저장경로

5-6

# 연습실 예약 플랫폼 사이트

<https://github.com/buchonsi/StudioRes>

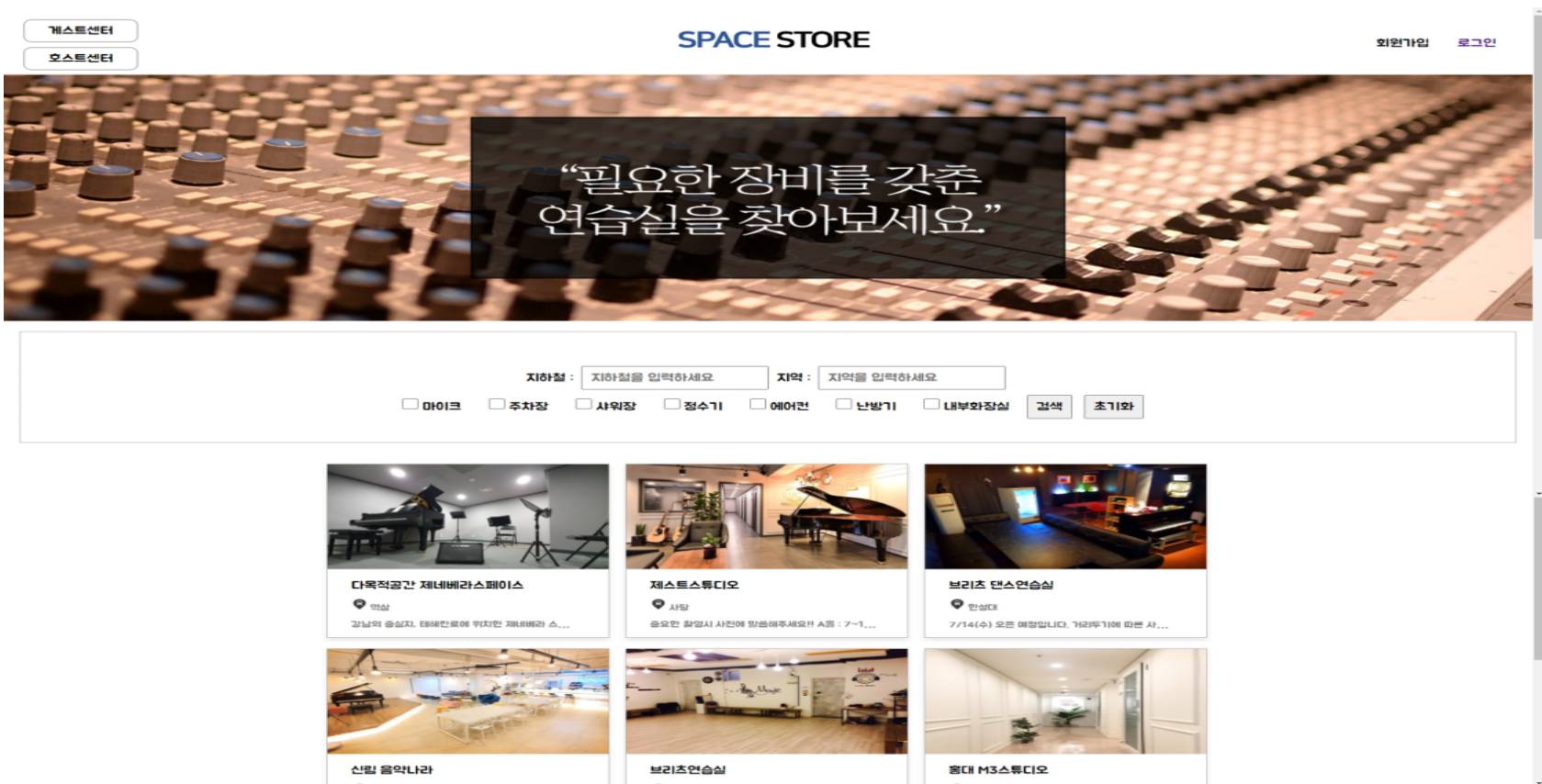
- 음악 연습실 대여, 회의실 대여, 공연장 대여 등 특정 공간 대여에 대한 수요가 늘어남에 따라 소비자가 쉽게 본인에게 필요한 공간을 찾고 예약할 수 있는 웹 페이지를 구현하고자 하였습니다.
- 사용자는 연습실을 조회하고 예약할 수 있습니다.
- 호스트는 연습실을 등록하고 연습실을 예약한 회원관리를 할 수 있습니다.
- 관리자는 페이지의 각종 정보를 조회하고 관리할 수 있습니다.

2021.06.01 - 2021.06.30 (4주)

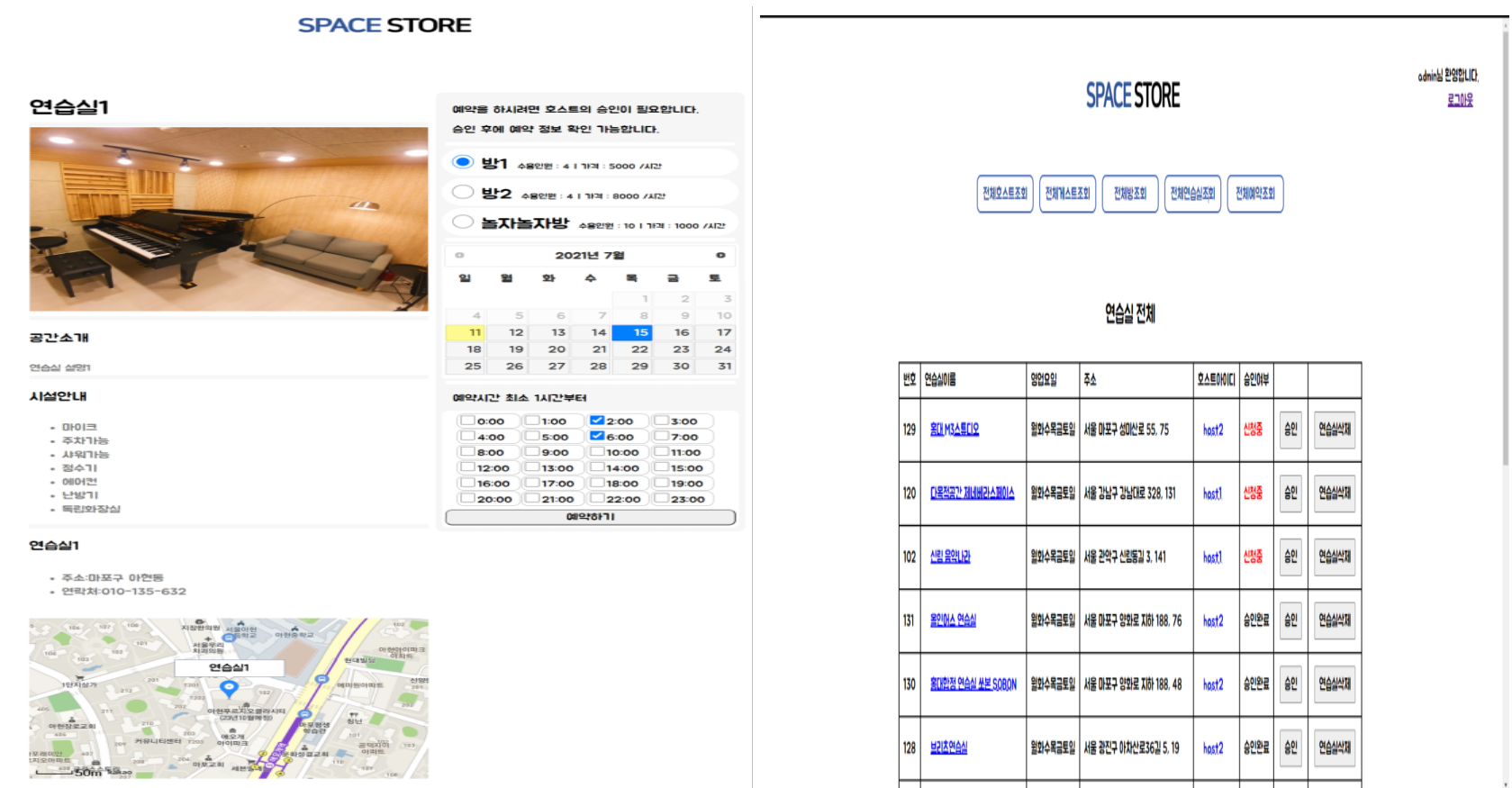
윤종무, 김현유, 최한길, 박태웅

## 사용 기술

- Java, MVC 모델
- Servlet
- Oracle
- JSP, jQuery, Ajax
- Github



<사용자 페이지>



<호스트/ 관리자 페이지>

## 연습실 예약 – Oracle Scheduler

--프로시저

```
create or replace procedure update_job_test
is
begin
update reservations set resv_check=2 where resv_date = TO_CHAR(SYSDATE , 'yyyy/mm/dd') and (resv_check=1 or resv_check=0);
commit;
end;
/
```

--스케줄러

```
begin
dbms_scheduler.create_job(
job_name => 'update_job_test1',
job_type => 'PLSQL_BLOCK',
job_action => 'BEGIN update_job_test;END;',
start_date => TO_TIMESTAMP_TZ('2021/04/26 23:50:00', 'yyyy/mm/dd hh24:mi:ss'),
repeat_interval => 'FREQ=DAILY'
);
end;
/
```

-----컨트롤러-----

```
begin dbms_scheduler.run_job('update_job_test1');end; --잡 시작
begin dbms_scheduler.enable('update_job_test1');end; --잡실행 (사용가능)

BEGIN dbms_scheduler.disable ('update_job_test1');END; -- 일시중지
BEGIN dbms_scheduler.enable ('update_job_test1');END; -- 다시시작
```

- - 예약상태(예약(0), 결제완료(1))를 사용완료(2)로 바꿔주는 프로시저를 생성합니다.
- - 잡스케줄러로 시간대(저녁 11:50)와 반복주기(24h)를 설정하여 해당시간에 작동하도록 하였습니다

## 연습실 예약 - Oracle Scheduler

SQL | 50개의 행이 인출됨(0.011초)

RESV_NO	GUEST...	ROOM_NO	RESV_D...	RESV_TIME	RESV_CHECK
28	122	102	107 21/05/04		1 0
29	123	102	107 21/05/04		2 0
30	124	102	107 21/05/04		3 0
31	125	102	105 21/05/03		8 0
32	126	102	105 21/05/03		9 0
33	127	102	105 21/05/03		10 0
34	128	102	105 21/05/03		11 0
35	129	102	102 21/05/03		17 0
36	130	102	102 21/05/03		18 0
37	131	102	102 21/05/03		19 0
38	132	102	102 21/05/03		20 0
39	134	102	127 21/05/18		21

메시지 - 로그 저장됨: C:\Users\WKOSTA\Desktop\Wbob\_insert(스케줄러추가).sql | 열 | 삽입 | Windows

오후 8:38  
2021-05-03

RESV_NO	GUEST...	ROOM_NO	RESV_D...	RESV_TIME	RESV_CHECK
28	122	102	107 21/05/04		1 0
29	123	102	107 21/05/04		2 0
30	124	102	107 21/05/04		3 0
31	125	102	105 21/05/03		8 2
32	126	102	105 21/05/03		9 2
33	127	102	105 21/05/03		10 2
34	128	102	105 21/05/03		11 2
35	129	102	102 21/05/03		17 2
36	130	102	102 21/05/03		18 2
37	131	102	102 21/05/03		19 2
38	132	102	102 21/05/03		20 2
39	134	102	127 21/05/18		21

메시지 - 로그 여러 누름(107) 저장됨(107) | 열 | 삽입 | Windows

오전 9:05  
2021-05-04

- 하루가 지나자 resv\_check가 0 or 1 에서 2로 바뀐것을 볼 수 있습니다.

## 연습실 예약 - 체크박스 데이터 처리

시설안내

마이크 ☐ 주차장 ☐ 샤워실 ☐ 정수기 ☐ 에어컨 ☐ 히터 ☐ 화장실 ☐

StudiolInsert.jsp

```
<fieldset>
  <legend>시설안내</legend>
  마이크<input type="checkbox" name="have" value="studio_have_mic">
  주차장<input type="checkbox" name="have" value="studio_have_park">
  샤워실<input type="checkbox" name="have" value="studio_have_shower">
  정수기<input type="checkbox" name="have" value="studio_have_water">
  에어컨<input type="checkbox" name="have" value="studio_have_aircon">
  히터<input type="checkbox" name="have" value="studio_have_heater">
  화장실<input type="checkbox" name="have" value="studio_have_toilet">
</fieldset>
```

배열 넘어감

```
//부대시설 체크
String[] facility = mutipartRequest.getParameterValues("have");
Map<String, String> facilityChk = new HashMap<String, String>();
facilityChk.put("studio_have_mic", "0");
facilityChk.put("studio_have_park", "0");
facilityChk.put("studio_have_shower", "0");
facilityChk.put("studio_have_water", "0");
facilityChk.put("studio_have_aircon", "0");
facilityChk.put("studio_have_heater", "0");
facilityChk.put("studio_have_toilet", "0");
```

- - (DB상에 0은 체크안함, 1은 체크)
- - HashMap에 키를 각 칼럼이름으로 값을 0으로 초기화해 줍니다.

```
if(facility != null) {
    for(String s:facility) {
        if(facilityChk.containsKey(s)) {
            facilityChk.replace(s, "1");
        }
    }
}
```

- Check 박스의 name을 같게 하고 submit을 하면 배열로 넘어옵니다.
- Value는 DB상의 칼럼이름으로 넣어줍니다.

- facilityChk.containsKey(s)
- facilityChk Map의 키값 중 s가 있는지 판단합니다  
Map으로 저장된 키값과 배열로 넘어온 값을 비교하여 같은 경우 값을 1로 고쳐줍니다.



연습실1



공간소개

연습실 설명1

시설안내

- 마이크
- 주차가능
- 샤워가능
- 정수기
- 에어컨
- 난방기
- 독립화장실

연습실1

- 주소:마포구 아현동
- 연락처:010-135-632



예약을 하시려면 호스트의 승인이 필요합니다.  
승인 후에 예약 정보 확인 가능합니다.

☒ 방1 수용인원 : 4 | 가격 : 5000 /시간

☐ 방2 수용인원 : 4 | 가격 : 8000 /시간

☐ 놀자놀자방 수용인원 : 10 | 가격 : 1000 /시간

2021년 7월						
일	월	화	수	목	금	토
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

예약시간 최소 1시간부터

- |                                |                                |  |                                |
|--------------------------------|--------------------------------|--|--------------------------------|
| <input type="checkbox"/> 0:00  | <input type="checkbox"/> 1:00  | <input checked="" type="checkbox"/> 2:00 | <input type="checkbox"/> 3:00  |
| <input type="checkbox"/> 4:00  | <input type="checkbox"/> 5:00  | <input checked="" type="checkbox"/> 6:00 | <input type="checkbox"/> 7:00  |
| <input type="checkbox"/> 8:00  | <input type="checkbox"/> 9:00  | <input type="checkbox"/> 10:00           | <input type="checkbox"/> 11:00 |
| <input type="checkbox"/> 12:00 | <input type="checkbox"/> 13:00 | <input type="checkbox"/> 14:00           | <input type="checkbox"/> 15:00 |
| <input type="checkbox"/> 16:00 | <input type="checkbox"/> 17:00 | <input type="checkbox"/> 18:00           | <input type="checkbox"/> 19:00 |
| <input type="checkbox"/> 20:00 | <input type="checkbox"/> 21:00 | <input type="checkbox"/> 22:00           | <input type="checkbox"/> 23:00 |

예약하기

연습실 예약 - 연습실 사진 저장/출력

- File 타입을 보낼때는 enctype="multipart/form-data" 를 사용합니다
- 파일이 저장되는 경로는 서버경로로 지정해 줍니다.

C:\eclipse-workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\StudioRes\imageUpload

<StudioInsert.jsp>

```
<form id="studiofrm" action="studioInsert" method="post" enctype="multipart/form-data">
  <fieldset>
    <legend>연습실 등록</legend>
    <label>연습실 명</label>
    <input type="text" class="input" name="studio_name"><br>
    <label>연습실 사진</label>
    <input type="file" name="studio_picture"><br>
```

<StudioInsertServlet.java>

```
//사진업로드
String upload_dir = "imageUpload";
int size = 1024*1024*10;
//서버경로
String path = getServletContext().getRealPath(upload_dir);

MultipartRequest mutipartRequest = new MultipartRequest(request, path, size, "utf-8",
    new DefaultFileRenamePolicy());
Enumeration files = mutipartRequest.getFileNames();
String str = (String)files.nextElement();
String originalFileName = mutipartRequest.getOriginalFileName(str);
```

- MultipartRequest를 생성하면 파일이 경로에 저장이 됩니다
- DB에는 파일이름을 저장해주기 위하여 파일명을 얻어왔습니다

15	연습실15	사진15	월화수목금토	\$
16	test1	%25B4%25F5_%25B1%25D7%25B7%25B9%25C0%25CC%25BD%25BA_%25BD%25BA%25C6%25A9%25B5%25F0%25BF%25C0_3.jpg	월화수목금토일	t
17	test12	01%25C8%25A6004.jpg	월화수목금토일	,

```
<div id="imagesizeDiv">
  <c:set var="pPath" value="${pageContext.request.contextPath}" />
  
</div>
```

- 이미지를 불러오기 위하여 경로와 파일 이름을 가져옵니다
- \${pageContext.request.contextPath} >> /StudioRes