



UI 구현

홍민준

2023-03-24 정기 수행평가

# 목차

- ◉ UI 설계 부분 - 스토리 보드
- ◉ UI 화면 구현
- ◉ 구현 코드

# Q 1 - UI 설계

/swiftER/er/er

Page	/search/er	화면이름	응급실 검색	설명	사용자 선택 조건에 맞는 응급실 검색 및 출력
url	http://swiftER/search/er				
<div><div>로고</div><div>마이페이지 회원정보</div><div>증상검색응급실검색약국검색커뮤니티고객센터</div><div><div>응급실 검색</div><div><div>1</div><div>위치 조건<div><div>부산</div><div>중구</div></div><div>진료 과목<div><div><input type="radio"/> 일반</div><div><input type="radio"/> 내과(중환자)</div><div><input type="radio"/> 외과(중환자)</div><div><input type="radio"/> 신경외과(중환자)</div></div><div>장비 정보<div><div><input type="radio"/> CT</div><div><input type="radio"/> MRI</div><div><input type="radio"/> 조영촬영기</div><div><input type="radio"/> 인큐베이터</div></div><div><div>초기화</div><div>2 검색</div></div></div></div></div></div></div></div>					<div><div>Description</div><div>1<div>검색 필터<ul style="list-style-type: none"><li>- 원하는 진료 과목, 장비 정보, 위치를 선택할 수 있도록 체크박스 출력(미선택시 모든 응급실 정보 출력)</li><li>- 위치 조건은 도시, 군/구 모두 드롭박스</li></ul></div></div><div>2<div>검색 및 초기화 버튼<ul style="list-style-type: none"><li>- 앱 테마에 맞는 색으로 지정</li><li>- 초기화 시 체크된 체크박스 모두 체크 해제</li><li>- 검색 버튼 클릭 시 하단에 검색 결과 출력</li></ul></div></div></div>

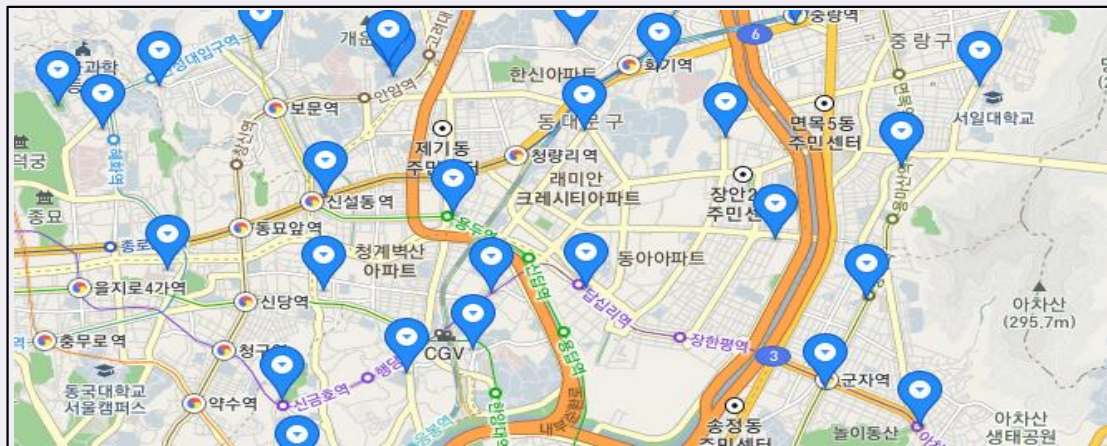
# Q 1 - UI 설계

/swiftER/er/er

Page	/search/er	화면이름	응급실 검색	설명	사용자 선택 조건에 맞는 응급실 검색 및 출력
url	http://swiftER/search/er				

선택하신 조건을 기반으로 검색한 결과입니다:

1



본 페이지에서 제공하는 내용은 참고사항일 뿐 게시물에 대한 법적책임은 없음을 밝혀드립니다.  
자세한 내용은 전문가와 상담하시기 바랍니다.

서비스소개   서비스이용약관   개인정보처리방침   사이트맵

로고

swiftER  
부산시 중구 광복로 93번길  
2023© All rights reserved. swiftER

Description

- 1 검색 결과 지도에 출력  
- 조건에 맞는 결과를 지도에 마커로 출력  
- 마커 마우스 호버 시 병원 이름 및 연락처 출력  
- 마커 클릭 시 병원 상세 페이지로 이동

# Q 1 - UI 설계

/swiftER/er/erDetail

Page	/search/erDetail	화면이름	응급실 개별 페이지	설명	사용자가 선택한 응급실 정보 및 평가 출력
url	http://swiftER/search/erDetail				
<div> <div>로고</div> <div>마이페이지 회원정보</div> <div> 증상검색 응급실검색 약국검색 커뮤니티 고객센터 </div> <div> <h2>한사랑 병원 응급실</h2> <div> <div> <div>위치</div> <div>부산광역시 중구 광복로 27번길</div> </div> <div> <div>진료과목</div> <div>안과, 내과, 정형외과</div> </div> <div> <div>공지사항</div> <div>*최근 1일 내 업데이트</div> <div>성형외과 수술 불가(당직의 부재)</div> </div> </div> <div> <div>연락처 정보</div> <div> <div>전화번호: 051-125-3464</div> <div>웹사이트: hansarang.org/</div> </div> </div> </div> </div>					<div>Description</div> <div> <div>1</div> <div> 응급실 정보 출력 <ul style="list-style-type: none"> <li>위치, 진료과목, 공지사항을 해당 공공 데이터 api에서 받아와서 출력</li> <li>탭이미지 css로 구현 불가 시 이미지 파일로 백그라운드에 넣기</li> </ul> </div> </div> <div> <div>2</div> <div> 연락처 정보 <ul style="list-style-type: none"> <li>전화번호(여러 개인 경우 모두 가져오기)와 웹사이트 정보를 해당 공공 데이터 api에서 받아와서 출력</li> </ul> </div> </div>

# Q 1 - UI 설계

/swiftER/er/erDetail

Page	/search/erDetail	화면이름	응급실 개별 페이지	설명	사용자가 선택한 응급실 정보 및 평가 출력
url	http://swiftER/search/erDetail				
<div>고객님의 의견을 들려주세요 1</div> <div><div><div>★★★★★</div><div>시설이 좋고 병원밥 맛은 여기가 최고입니다</div></div><div><div>★★★★★</div><div>코로나 확진자라고 한겨울에 칼바람 맞으면서 밖에서 대기하게 하더라고요. 실망</div></div><div><div>★★★★★</div><div>코로나 확진자라고 한겨울에 칼바람 맞으면서 밖에서 대기하게 하더라고요. 실망</div></div></div> <div>abc**** / 내원일 2023.02.28</div> <div>abc**** / 내원일 2023.02.28</div> <div>abc**** / 내원일 2023.02.28</div>					<div>Description</div> <div>1리뷰 출력<ul style="list-style-type: none"><li>- 해당 병원 리뷰 및 별점을 모두 출력</li><li>- 아이디 마스킹 처리</li><li>- 작성 날짜 출력</li></ul></div> <div>리뷰 작성은 마이 페이지 &gt; 병원 내원 기록에서 병원을 선택해서 남기기</div>
<div>서비스소개 서비스이용약관 개인정보처리방침 사이트맵</div>					
<div>로고 swiftER 부산시 중구 광복로 93번길 2023© All rights reserved. swiftER</div>					

# Q 2 - UI 구현 화면

/swiftER/er/er

응급실 검색

← → ↻ 127.0.0.1:8101/swiftER/er/er

로그인 | 회원가입

swiftER

응상검색 | 응급실검색 | 약국검색 | 커뮤니티 | 고객센터

응급실 검색

위치 조건  
전체

응급 수술(해당 수술 가능 병원)  
☐ 뇌종양수술 ☐ 뇌경색의재관류 ☐ 심근경색의재관류 ☐ 조산산모

장비 정보(가용여부)  
☐ CT ☐ MRI ☐ 초영광영상기 ☐ 인공호흡기 ☐ 인큐베이터

추가하기 | 검색

선택하신 조건을 기반으로 한 검색 결과 입니다.

본 페이지에서 제공하는 내용은 참고사항일 뿐 게시물에 대한 법적책임은 없음을 밝혀드립니다.  
자세한 내용은 전문가와 상담하시기 바랍니다.

서비스소개 | 서비스이용약관 | 개인정보처리방침 | 사이트맵

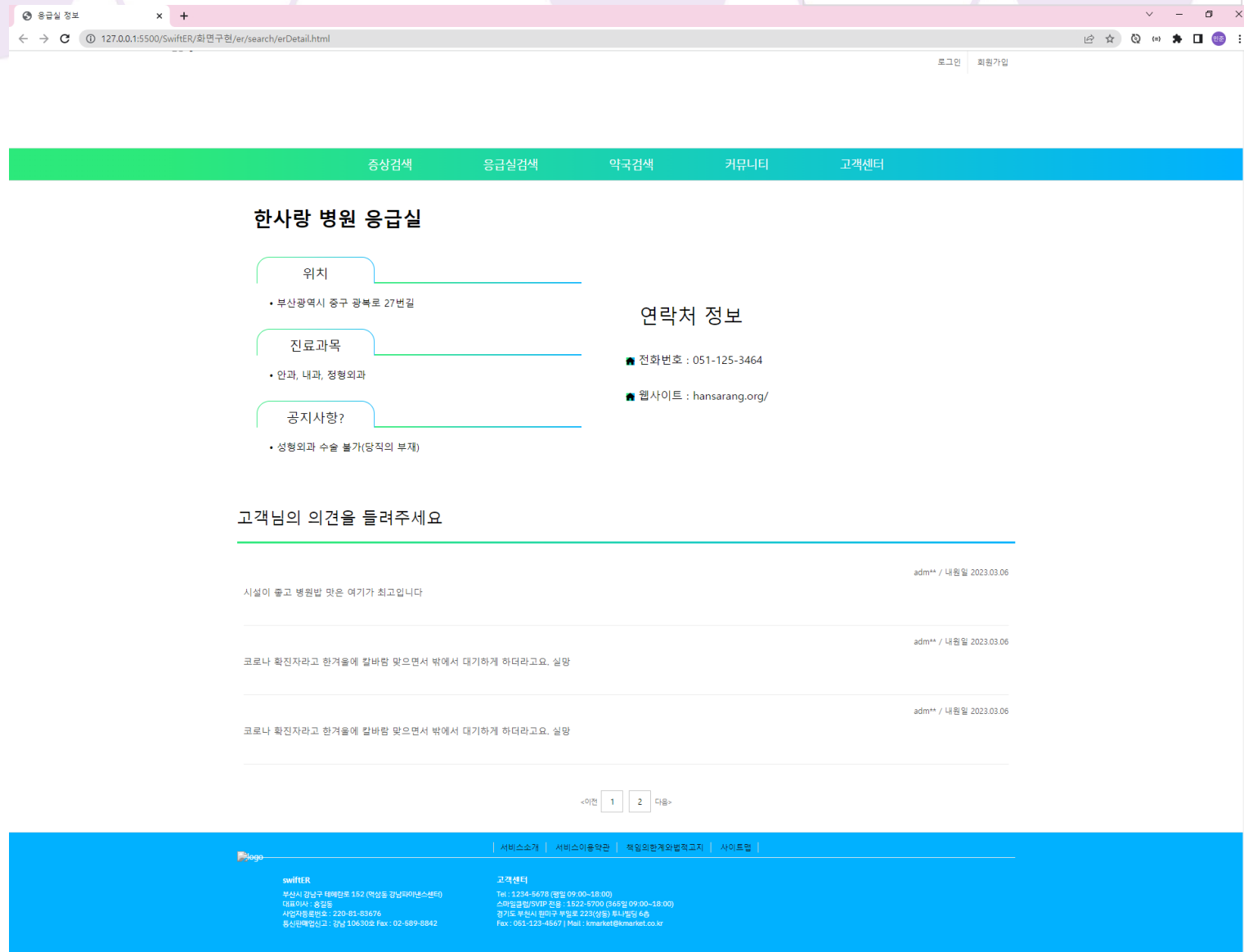
swiftER

주소: 경상구 새마을로 152 (의성읍) 054-810-5500  
대표이사: 윤종도  
사업자등록번호: 220-81-83676  
통신판매업신고: 영남 104-308 | Fax: 02-569-8842

고객센터  
Tel: 1234-5678 (월~금 09:00~18:00)  
스마트폰 앱 다운로드: 1522-8700 (월~금 09:00~18:00)  
3월~5월: 051-51071 | 7월~9월: 221-0101 | 8월~9월: 84-7551  
Fax: 051-523-4567 | Mail: internet@kmarket.co.kr

# Q 2 - UI 구현 화면

/swiftER/er/erDetail





# Q 3 - 구현 코드(er)

## HTML

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org"
3     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
4 <head>
5     <meta charset="UTF-8">
6     <title>응급실 검색</title>
7     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
8     <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
9     <link rel="stylesheet" href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/themes/smoothness/jquery-ui.css">
10    <link rel="preconnect" href="https://fonts.googleapis.com">
11    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12    <link href="https://fonts.googleapis.com/css2?family=IBM+Plex+Sans+KR&display=swap" rel="stylesheet">
13    <link rel="stylesheet" href="/swifter/css/common.css"/>
14    <link rel="stylesheet" href="/swifter/css/er.css"/>
15    <link rel="stylesheet" href="/swifter/css/er.css"/>
16 </head>
17 <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=4b25c276d476675dffb5920e680cf36"></script>
18 <body>
19     <script src="/swifter/js/er.js"></script>
20     <div id="wrapper">
21         <header>
22             <div class="top">
23                 <div>
24                     <a href="#">로그인</a>
25                     <a href="#">회원가입</a>
26                 </div>
27             </div>
28             <div class="logo">
29                 <div>
30                     <a href="#">
31                         
32                     </a>
33                 </div>
34             </div>
35             <div class="menu">
36                 <div>
37                     <ul>
38                         <li>
39                             <a href="#">중앙검색</a>
40                         </li>
41                         <li>
42                             <a href="#">응급실검색</a>
43                         </li>
44                         <li>
45                             <a href="#">약국검색</a>
46                         </li>
47                         <li>
48                             <a href="#">커뮤니티</a>
49                         </li>
50                         <li>
51                             <a href="#">고객센터</a>
52                         </li>
53                     </ul>
54                 </div>
55             </div>
56         </header>
57         <main id="search">
58             <section class="er">
59                 <div class="title">
60                     <h2 class="er-search">응급실 검색</h2>
61                 </div>
62                 <div class="selectBox">
63                     <form action="#" method="post">
64                         <div>
65                             <fieldset class="location">
66                                 <legend>위치 조건</legend>
67                                 <select name="city">
68                                     <option value="전체">전체</option>
69                                     <th:block th:each="region, i:${region}">
```

```
70                     <option th:value="${region.region}" id="region">[[${region.region}]]</option>
71                 </th:block>
72             </select>
73             <select name="town">
74             </select>
75         </fieldset>
76         <fieldset class="med-department">
77             <legend>응급 수술 (해당 수술 가능 병원)</legend>
78             <input type="checkbox" name="일반" value="common"><label for="common">뇌출혈수술</label>
79             <input type="checkbox" name="내과" value="medicine"><label for="medicine">뇌경색의재관류</label>
80             <input type="checkbox" name="외과" value="surgery"><label for="surgery">심근경색의재관류</label>
81             <input type="checkbox" name="신경외과" value="neurosurgery"><label for="neurosurgery">조산산모</label>
82         </fieldset>
83         <fieldset class="med-equipment">
84             <legend>장비 정보 (가용여부)</legend>
85             <input type="checkbox" name="ct" value="ct"><label for="ct">CT</label>
86             <input type="checkbox" name="mri" value="mri"><label for="mri">MRI</label>
87             <input type="checkbox" name="조영촬영기" value="camera"><label for="camera">조영촬영기</label>
88             <input type="checkbox" name="인공호흡기" value="respirator"><label for="respirator">인공호흡기</label>
89             <input type="checkbox" name="인큐베이터" value="incubator"><label for="incubator">인큐베이터</label>
90         </fieldset>
91     </div>
92     <div class="btn">
93         <input type="reset" value="초기화" class="resetBtn">
94         <input type="button" value="검색" class="subBtn">
95     </div>
96 </form>
97 </div>
98 <div class="search_map">
99     <h2>선택하신 조건을 기반으로 한 검색 결과 입니다.<span class="totalER"></span></h2>
100    <div id="map">
101    </div>
102 </div>
103 <div class="info">
104    <p>
105        본 페이지에서 제공하는 내용은 참고사항일 뿐 게시물에 대한 법적책임은 없음을 밝혀드립니다. <br>
106        자세한 내용은 전문가와 상담하시기 바랍니다.
107    </p>
108 </div>
109 </section>
110 </main>
111 <th:block th:insert="~{[_footer.html]}">
```

# Q 3 - 구현 코드(er)

JS

```
1 $(document).ready(function () {
2   let er = [];
3   let erChk = [];
4   let erChk2 = [];
5   let combined = [];
6   let combinedFiltered = [];
7
8   var container = document.getElementById('map'); //지도도를 담을 영역의 DOM 레퍼런스
9   var options = { //지도를 생성할 때 필요한 기본 옵션
10     center: new kakao.maps.LatLng(33.450781, 126.570667), //지도의 중심좌표.
11     level: 3 //지도의 레벨(확대, 축소 정도)
12   };
13
14   var map = new kakao.maps.Map(container, options); //지도 생성 및 객체 리턴
15
16   $('select[name=city]').on('change', function(){
17     let city = $(this).val();
18     subregion(city);
19   });
20
21   $('.subBtn').click(function(e){
22     e.preventDefault();
23
24     let city = $('select[name=city]').val();
25     let town = $('select[name=town]').val();
26
27     let jsonData = {
28       "city": city,
29       "town": town
30     };
31
32     console.log('data', jsonData);
33
34     $.ajax({
35       url: '/swifter/er/erSearch',
36       method: 'POST',
37       data: jsonData,
38       contentType: 'application/x-www-form-urlencoded; charset=UTF-8',
39       success: function(data){
40         let json = JSON.parse(data);
41         let check = json.response.body.totalCount;
42         if(check == 0){
43           alert("검색한 결과가 없습니다, 다른 조건으로 검색하시길 바랍니다.");
44           location.reload();
45         }
46
47         let items = json.response.body.items.item;
48         er = items;
49         console.log('er', er);
50
51         $.ajax({
52           url: '/swifter/er/erSearch2',
53           async: false,
54           method: 'POST',
55           data: jsonData,
56           contentType: 'application/x-www-form-urlencoded; charset=UTF-8',
57           success: function(data){
58             let json = JSON.parse(data);
59             let check = json.response.body.totalCount;
60             if(check == 0){
61               alert("검색한 결과가 없습니다, 다른 조건으로 검색하시길 바랍니다.");
62               location.reload();
63             }
64
65             let items = json.response.body.items.item;
66             erChk = items;
67           }
68         });
69   });
```

```
70     $.ajax({
71       url: '/swifter/er/erSearch3',
72       async: false,
73       method: 'POST',
74       data: jsonData,
75       contentType: 'application/x-www-form-urlencoded; charset=UTF-8',
76       success: function(data){
77         let json = JSON.parse(data);
78         let check = json.response.body.totalCount;
79         if(check == 0){
80           alert("검색한 결과가 없습니다, 다른 조건으로 검색하시길 바랍니다.");
81           location.reload();
82         }
83
84         let items = json.response.body.items.item;
85         erChk2 = items;
86       });
87
88   let common = $('input[name=일반]').prop("checked");
89   let medicine = $('input[name=내과]').prop("checked");
90   let surgery = $('input[name=외과]').prop("checked");
91   let neurosurgery = $('input[name=신경외과]').prop("checked");
92   console.log('common', common);
93   console.log('medicine', medicine);
94   console.log('surgery', surgery);
95   console.log('neurosurgery', neurosurgery);
96
97   let ct = $('input[name=ct]').prop("checked");
98   let mri = $('input[name=mri]').prop("checked");
99   let camera = $('input[name=초음파장기]').prop("checked");
100  let respirator = $('input[name=인공호흡기]').prop("checked");
101  let incubator = $('input[name=인큐베이터]').prop("checked");
102  console.log('ct', ct);
103  console.log('mri', mri);
104  console.log('camera', camera);
105  console.log('respirator', respirator);
106  console.log('incubator', incubator);
107
108  // 배열 합치기
109  erChk.sort((a, b) => a.hpid.localeCompare(b.hpid));
110  erChk2.sort((a, b) => a.hpid.localeCompare(b.hpid));
111
112  let mergedArray = erChk.map((obj, index) => {
113    return obj.hpid == erChk2[index].hpid
114      ? Object.assign({}, obj, erChk2[index])
115      : obj;
116  });
117
118  er.slice(erChk.length).forEach(obj => mergedArray.push(obj));
119
120  let erMerged = mergedArray.map(obj => {
121    return {
122      dutyAddr: obj.dutyAddr,
123      dutyEmcls: obj.dutyEmcls,
124      dutyEmclsName: obj.dutyEmclsName,
125      dutyName: obj.dutyName,
126      dutyTel1: obj.dutyTel1,
127      dutyTel2: obj.dutyTel2,
128      hpid: obj.hpid,
129      hv11: obj.hv11,
130      hvctayn: obj.hvctayn,
131      hvmlayn: obj.hvmlayn,
132      hvangioayn: obj.hvangioayn,
133      hvventiayn: obj.hvventiayn,
134      wgs84Lat: obj.wgs84Lat,
135      wgs84Lon: obj.wgs84Lon
136    };
137  });
138  });
```

```
140 console.log('erMerged', erMerged);
141 console.log('erChk2', erChk2);
142
143 // 검색 옵션을 선택한 경우
144 if(ct || mri || camera || respirator || incubator ||
145    common || medicine || surgery || neurosurgery){
146   combined = erMerged.map((erItem) => {
147     let matchingChk = erChk2.find((chkItem) => chkItem.dutyName === erItem.hpid);
148     if (matchingChk) {
149       return {
150         ...erItem,
151         matchingChk,
152         dutyName: erItem.dutyName
153       };
154     }
155     return null;
156   }).filter(item => item !== null); // null 값을 제외하는 필터링 추가
157
158   console.log('combined', combined);
159
160   let filters = [];
161
162   if (common) {
163     filters.push((arr) => {
164       console.log('mkioskTy1 filter arr:', arr); // arr 출력
165       return arr.filter(item => item['mkioskTy1'] == "Y");
166     });
167   }
168   if (medicine) {
169     filters.push((arr) => arr.filter(item => item['mkioskTy2'] == "Y"));
170   }
171   if (surgery) {
172     filters.push((arr) => {
173       console.log('mkioskTy3 filter arr:', arr); // arr 출력
174       return arr.filter(item => item['mkioskTy3'] == "Y");
175     });
176   }
177   if (neurosurgery) {
178     filters.push((arr) => arr.filter(item => item['mkioskTy8'] == "Y"));
179   }
180   if (ct) {
181     filters.push((arr) => {
182       console.log('hvctayn filter arr:', arr); // arr 출력
183       return arr.filter(item => item['hvctayn'] == "Y");
184     });
185   }
186   if (mri) {
187     filters.push((arr) => arr.filter(item => item['hvmlayn'] == "Y"));
188   }
189   if (camera) {
190     filters.push((arr) => arr.filter(item => item['hvngioayn'] == "Y"));
191   }
192   if (respirator) {
193     filters.push((arr) => arr.filter(item => item['hvventiayn'] == "Y"));
194   }
195   if (incubator) {
196     filters.push((arr) => arr.filter(item => item['hvii'] > 0));
197   }
198
199   let filtered = combined;
200
201   if (filters.length > 0) {
202     filters.forEach(filterFn => {
203       filtered = filterFn(filtered);
204     });
205   }
206
207 }
```

# Q 3 - 구현 코드(er)

JS

```
208     if(filtered.length > 0){
209         alert('선택하신 조건을 모두 충족하는 결과가 없습니다, 선택하신 지역을 기반으로 한 검색 결과입니다.');
```

```
210         combinedFiltered = combined;
211     }else{
212         combinedFiltered = filtered;
213     }
214
215     console.log('filters', filters);
216     console.log('filtered', filtered);
217
218     console.log('combinedFiltered', combinedFiltered);
219 }else{
220     // 지역만 선택한 경우
221     combinedFiltered = erMerged;
222 }
223
224 // 마커 이미지의 이미지 주소입니다
225 var imageSrc = "https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerStar.png";
226
227 // 마커 이미지의 이미지 크기 입니다
228 var imageSize = new kakao.maps.Size(24, 35);
229 var latlng = new kakao.maps.LatLng(combinedFiltered[0].wgs84Lat, combinedFiltered[0].wgs84Lon);
230 combinedFiltered.forEach(function (item) {
231     // 마커 이미지를 생성합니다
232     var markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize);
233
234     // 마커를 생성합니다
235     var marker = new kakao.maps.Marker({
236         map: map, // 마커를 표시할 지도
237         position: new kakao.maps.LatLng(item.wgs84Lat, item.wgs84Lon),
238         image: markerImage // 마커 이미지
239     });
240
241     //이동할 위도 경도 위치를 생성합니다
242     var moveLatLon = latlng;
243     // 지도 중심을 부드럽게 이동시킵니다
244     // 만약 이동할 거리가 지도 화면보다 크면 부드러운 효과 없이 이동합니다
245     map.panto(moveLatLon);
246
247     // 마커를 중심으로 커스텀 오버레이를 표시하기위해 CSS를 이용해 위치를 설정했습니다
248     var overlay = new kakao.maps.CustomOverlay({
249         position: new kakao.maps.LatLng(item.wgs84Lat, item.wgs84Lon)
250     });
251
252     // widowInfo 태그 생성
253     var content = document.createElement('div');
254     content.className = 'wrap';
255
256     var info = document.createElement('div');
257     info.className = 'info';
258
259     var title = document.createElement('div');
260     title.className = 'title';
261     title.innerHTML = 'title';
262     title.innerHTML = item.dutyName;
263
264     var close = document.createElement('div');
265     close.className = 'close';
266     close.onclick = function() {overlay.setMap(null)};
267     close.title = '닫기';
268
269     var body = document.createElement('div');
270     body.className = 'body';
271
272     var desc = document.createElement('div');
273     desc.className = 'desc';
274
275     var ellipsis = document.createElement('div');
276     ellipsis.className = 'ellipsis';
```

```
277     ellipsis.innerHTML = item.dutyAddr;
278
279     var jibun = document.createElement('div');
280     jibun.className = 'jibun';
281     jibun.innerHTML = '연락처1 : '+item.dutyTel1+' 연락처2 : '+item.dutyTel3;
282
283     var linkdiv = document.createElement('div');
284
285     var code = document.createElement('input');
286     code.type = 'hidden';
287     code.name = 'code';
288     code.value = item.hpId;
289     linkdiv.appendChild(code);
290
291     var link = document.createElement('a');
292     link.className = 'link';
293     link.href = '/swiftER/er/erDetail';
294     link.innerHTML = '자세히 보기';
295
296     linkdiv.appendChild(link);
297     linkdiv.appendChild(code);
298     desc.appendChild(ellipsis);
299     desc.appendChild(jibun);
300     desc.appendChild(linkdiv);
301     body.appendChild(desc);
302     title.appendChild(close);
303     info.appendChild(title);
304     info.appendChild(body);
305     content.appendChild(info);
306
307     console.log(content)
308     // widowInfo 태그 생성 끝
309
310     overlay.setContent(content);
311     //overlay.setMap(map);
312
313     // 마커를 클릭했을 때 커스텀 오버레이를 표시합니다
314     kakao.maps.event.addListener(marker, 'click', function() {
315         console.log('marker : %o', marker);
316         overlay.setMap(map);
317         var hpId = $('input[name=code]').val();
318         var hf = '/swiftER/er/erDetail?code='+hpId;
319
320         $('link').attr("href", hf);
321     });
322
323     });
324
325     });
326 });
327
328 function subregion(city){
329     let jsonData = {"city":city}
330     $.ajax({
331         url: '/swiftER/er/subregion',
332         method: 'POST',
333         dataType: 'json',
334         data: jsonData,
335         success: (data) => {
336             let subregion = $('select[name=town']');
337             subregion.children().remove();
338             let tag = '<option value="">전체</option>';
339             data.forEach(async data => {
340                 tag += '<option th:value="'+data.subregion">th:text="'+data.subregion">'+data.subregion</option>';
341             });
342             subregion.append(tag);
343         }
344     });
345 }
```

# Q 3 - 구현 코드(er)

## ERRestController

```
20 import java.io.IOException;
21 @Controller
22 public class ERRestController {
23
24     @Autowired
25     private ERService service;
26
27     @Value("${restApi.key}")
28     private String restApiKey;
29
30     @Value("${restApi.erdetailurl}")
31     private String erdetailurl;
32
33     @Value("${restApi.erlisturl}")
34     private String erlisturl;
35
36     @Value("${restApi.eruserfulurl}")
37     private String eruserfulurl;
38
39     @Value("${restApi.ersmtypeurl}")
40     private String ersmtypeurl;
41
42     private String pageNo = "1";
43     private String numofRows = "1000";
44
45     @ResponseBody
46     @PostMapping("er/erSearch")
47     public String erSearch(String city, String town) throws IOException {
48         // RestTemplate 생성
49         RestTemplate restTemplate = new RestTemplate();
50
51         // 오브젝트로 결과값 받아오기
52         String url = erlisturl + restApiKey + "&q=" + city + "&q1=" + town + "&pageNo=" + pageNo + "&numOfRows=" + numofRows;
53         System.out.println("url : " + url);
54         String response = restTemplate.getForObject(url, String.class);
55
56         return response;
57     }
58
59     @ResponseBody
60     @PostMapping("er/erSearch2")
61     public String erSearch2(String city, String town) throws IOException {
62         // RestTemplate 생성
63         RestTemplate restTemplate = new RestTemplate();
64
65         // 오브젝트로 결과값 받아오기
66         String url = eruserfulurl + restApiKey + "&STAGE1=" + city + "&STAGE2=" + town + "&pageNo=" + pageNo + "&numOfRows=" + numofRows;
67         System.out.println("url : " + url);
68         String response = restTemplate.getForObject(url, String.class);
69
70         return response;
71     }
72
73     @ResponseBody
74     @PostMapping("er/erSearch3")
75     public String erSearch3(String city, String town) throws IOException {
76         // RestTemplate 생성
77         RestTemplate restTemplate = new RestTemplate();
78
79         // 오브젝트로 결과값 받아오기
80         String url = ersmtypeurl + restApiKey + "&STAGE1=" + city + "&STAGE2=" + town + "&pageNo=" + pageNo + "&numOfRows=" + numofRows;
81         System.out.println("url : " + url);
82         String response = restTemplate.getForObject(url, String.class);
83
84     }
```

```
85     return response;
86 }
87
88 @GetMapping("er/erDetail")
89 public String erDetail(Model model, @RequestParam("code") String code) {
90     System.out.println("code : " + code);
91
92     List<ERReviewVO> reviews = service.selectErReview(code);
93     model.addAttribute("reviews", reviews);
94     model.addAttribute("code", code);
95
96     return "er/erDetail";
97 }
98
99 @ResponseBody
100 @PostMapping("er/erDetailInfo")
101 public String erDetailInfo(@RequestParam("code") String code) throws IOException {
102     String deurl = erdetailurl;
103     System.out.println("code : " + code);
104
105     // RestTemplate 생성
106     RestTemplate restTemplate = new RestTemplate();
107
108     // 오브젝트로 결과값 받아오기
109     String url = deurl + restApiKey + "&HPID=" + code + "&pageNo=" + pageNo + "&numOfRows=" + numofRows;
110     System.out.println("url : " + url);
111     String response = restTemplate.getForObject(url, String.class);
112
113     return response;
114 }
115
116 }
```

## ERController

```
1 package kr.co.swiftER.controller;
2
3 import java.util.List;
4
5 @Controller
6 public class ERController {
7
8     @Autowired
9     private ERService service;
10
11     @GetMapping("er/er")
12     public String er(Model model) {
13         List<ERCateVO> region = service.selectErCate();
14
15         model.addAttribute("region", region);
16
17         return "er/er";
18     }
19
20     @PostMapping("er/subregion")
21     @ResponseBody
22     public List<ERSubcateVO> cate(String city) {
23         System.out.println("city : " + city);
24
25         List<ERSubcateVO> sub = service.selectErSubCate(city);
26
27         return sub;
28     }
29 }
```

## ERDAO

```
1 package kr.co.swiftER.dao;
2
3 import java.util.List;
4
5 @Mapper
6 @Repository
7 public interface ERDAO {
8
9     public List<ERCateVO> selectErCate();
10     public List<ERSubcateVO> selectErSubCate(String city);
11     public List<ERReviewVO> selectErReview(String code);
12
13 }
```

# Q 3 - 구현 코드(er)

## ERService

```
1 package kr.co.swifter.service;
2
3 import java.util.List;
4
5 @Service
6 public class ERSERVICE {
7
8     @Autowired
9     private ERDAO dao;
10
11     public List<ERCateVO> selectERCate(){
12         return dao.selectERCate();
13     }
14
15     public List<ERSubcateVO> selectERSubCate(String city){
16         return dao.selectERSubCate(city);
17     }
18
19     public List<ERReviewVO> selectERReview(String code) {
20         return dao.selectERReview(code);
21     }
22
23 }
```

## ERReviewVO

```
1 package kr.co.swifter.vo;
2
3 import lombok.AllArgsConstructor;
4
5 @Data
6 @AllArgsConstructor
7 @NoArgsConstructor
8 public class ERReviewVO {
9     private int no;
10    private String member_uid;
11    private String hospital_code;
12    private int region_code;
13    private int subregion_code;
14    private String title;
15    private int rating;
16    private String content;
17    private String regip;
18    private String rdate;
19
20    //추가
21    private String region;
22    private String subregion;
23 }
```

## ERCateVO

```
1 package kr.co.swifter.vo;
2
3 import lombok.Data;
4
5 @Data
6 public class ERCateVO {
7     private String region;
8     private int region_code;
9
10 }
```

## ERSubCateVO

```
1 package kr.co.swifter.vo;
2
3 import lombok.Data;
4
5 @Data
6 public class ERSubcateVO {
7     private String subregion;
8     private int region_code;
9     private int subregion_code;
10
11 }
```

## ermapper

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper
3     PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
4     "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="kr.co.swifter.dao.ERDAO">
6
7     <insert id="insertArticle" useGeneratedKeys="true" keyProperty="no">
8
9     </insert>
10
11     <select id="selectERCate" resultType="kr.co.swifter.vo.ERCateVO">
12         select * from `search_er_cate`;
13     </select>
14
15     <select id="selectERSubCate" resultType="kr.co.swifter.vo.ERSubcateVO">
16         select * from `search_er_subcate` where `region` = #{city};
17     </select>
18
19     <select id="selectERReview" resultType="kr.co.swifter.vo.ERReviewVO">
20         select * from `search_er_review` where `hospital_code` = #{code};
21     </select>
22 </mapper>
```

# Q 3 - 구현 코드(erDetail)

## erDetail

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org"
3     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
4 <head>
5     <meta charset="UTF-8">
6     <title>음습실 정보</title>
7     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
8     <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
9     <link rel="preconnect" href="https://fonts.googleapis.com">
10    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
11    <link href="https://fonts.googleapis.com/css2?family=IBM+Plex+Sans+KR&display=swap" rel="stylesheet">
12    <script src="https://kit.fontawesome.com/e8f010863.js" crossorigin="anonymous"></script>
13    <link rel="stylesheet" href="/swiftER/css/common.css"/>
14    <link rel="stylesheet" href="/swiftER/css/er.css"/>
15 </head>
16 <body>
17     <div id="wrapper">
18         <header>
19             <div class="top">
20                 <div>
21                     <a href="#">로그인</a>
22                     <a href="#">회원가입</a>
23                 </div>
24                 <div class="logo">
25                     <div>
26                         <a href="#">
27                             
28                         </a>
29                     </div>
30                 </div>
31             </div>
32             <div class="menu">
33                 <div>
34                     <ul>
35                         <li>
36                             <a href="#">중앙검색</a>
37                         </li>
38                         <li>
39                             <a href="#">음습실검색</a>
40                         </li>
41                         <li>
42                             <a href="#">약국검색</a>
43                         </li>
44                         <li>
45                             <a href="#">커뮤니티</a>
46                         </li>
47                         <li>
48                             <a href="#">고객센터</a>
49                         </li>
50                     </ul>
51                 </div>
52             </div>
53         </header>
54         <div id="search">
55             <section class="Detail">
56                 <div class="ErInfo">
57                     <nav>
58                         <h1 id="hosp_name"></h1>
59                     </nav>
60                     <article class="part1">
61                         <input type="hidden" name="hpid" th:value="${code}"/>
62                         <div class="border1">
63                             <h2>위지</h2>
64                         </div>
65                         <div class="border2">
66                             <p id="addr">

```

```
70     </p>
71     </article>
72     <article>
73         <div class="border1"><h2>진료과목</h2></div>
74         <div class="border2">
75             <p id="dep">
76             </p>
77         </div>
78     </article>
79     <article>
80         <div class="border1"><h2>찾아오시는 길</h2></div>
81         <div class="border2">
82             <p id="etc">
83             </p>
84         </div>
85     </article>
86     <article class="part2">
87         <div>
88             <nav><h2>연락처 정보</h2></nav>
89             <div>
90                 <span id="tel"><i class="fas fa-home"></i></span><br>
91                 <span id="etc1"><i class="fas fa-home"></i></span>
92             </div>
93         </div>
94     </article>
95     <article class="review">
96         <nav><h1>고객님의 의견을 들려주세요</h1></nav>
97         <div>
98             <th:block th:each="review : ${reviews}">
99                 <div>
100                     <h5 class="rating star" th:attrappend="class= '${review.rating}'">별점</h5>
101                     <span>[[${review.member_uid}]] / 작성일 [[${review.rdate}]]</span>
102                 </div>
103                 <p th:text="${review.content}">
104                 </p>
105             </li>
106         </th:block>
107     </div>
108     <div class="paging">
109         <span class="prev"><a href="#">&lt;</a></span>
110         <span class="num">
111             <a href="#" class="on">1</a>
112             <a href="#">2</a>
113         </span>
114         <span class="prev"><a href="#">다음</a></span>
115     </div>
116 </div>
117 </section>
118 </div>
119 </main>
```

## JS

```
1 $(document).ready(function () {
2     let code = $('input[name=hpid]').val();
3     let detail = [];
4
5     let jsonData = {
6         "code": code
7     };
8
9     console.log('jsonData', jsonData);
10    $.ajax({
11        url: "/swiftER/er/erDetailInfo",
12        method: "POST",
13        data: jsonData,
14        success: function (data) {
15            console.log('data : ', data);
16            let json = JSON.parse(data);
17            let check = json.response.body.totalCount;
18            if (check == 0) {
19                alert("검색한 결과가 없습니다.");
20                location.href = "/swiftER/er/er";
21            }
22
23            let items = json.response.body.items.item;
24            detail = items;
25            console.log('detail', detail);
26
27            let name = document.getElementById('hosp_name');
28            let addr = document.getElementById('addr');
29            let dep = document.getElementById('dep');
30            let info = document.getElementById('etc');
31            let tel = document.getElementById('tel');
32            let etc1 = document.getElementById('etc1');
33
34            name.innerText = ' ' + detail.dutyName;
35            addr.innerText = ' ' + detail.dutyAddr;
36            dep.innerText = ' ' + detail.dgidName;
37            info.innerText = ' ' + detail.dutyMapping;
38            tel.innerText = '음습실 전화 : ' + detail.dutyTel3;
39            etc1.innerText = '내용이 없습니다.';
40
41        }
42    });
43
44
45
46
47 });
```