N-EMER

Nearby EMERgency

Final Project Report

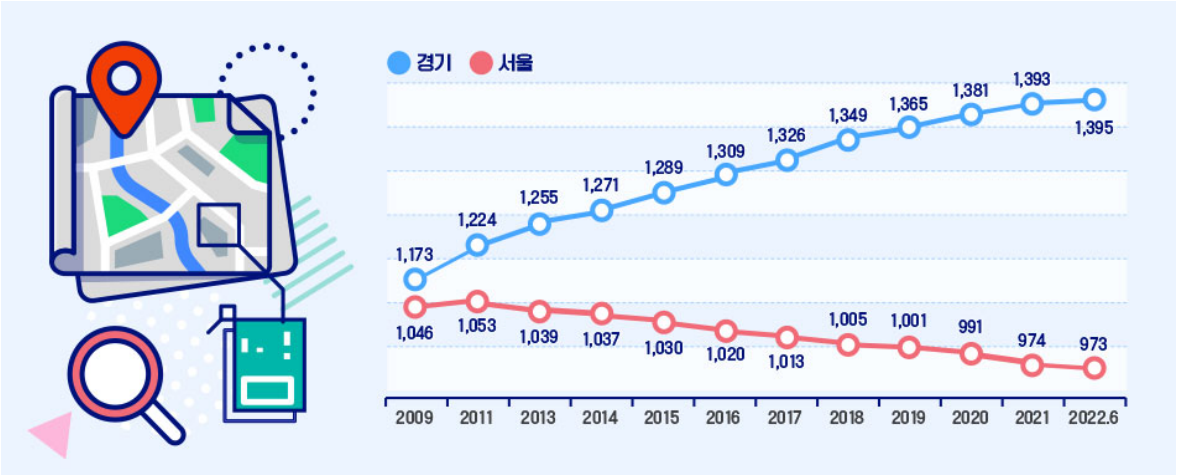
- Web Programming Lab 2022 -

Team N-EMER

Student : 2018311492 Jaesung Lee

Student : 2018312113 Jaemin You

Project Name
N-EMER
Project Overview



As many as 13.948 million people live in Gyeonggi-do, where we are now. It's a huge figure. If you look at the statistical graph, you can see that the figure is increasing.

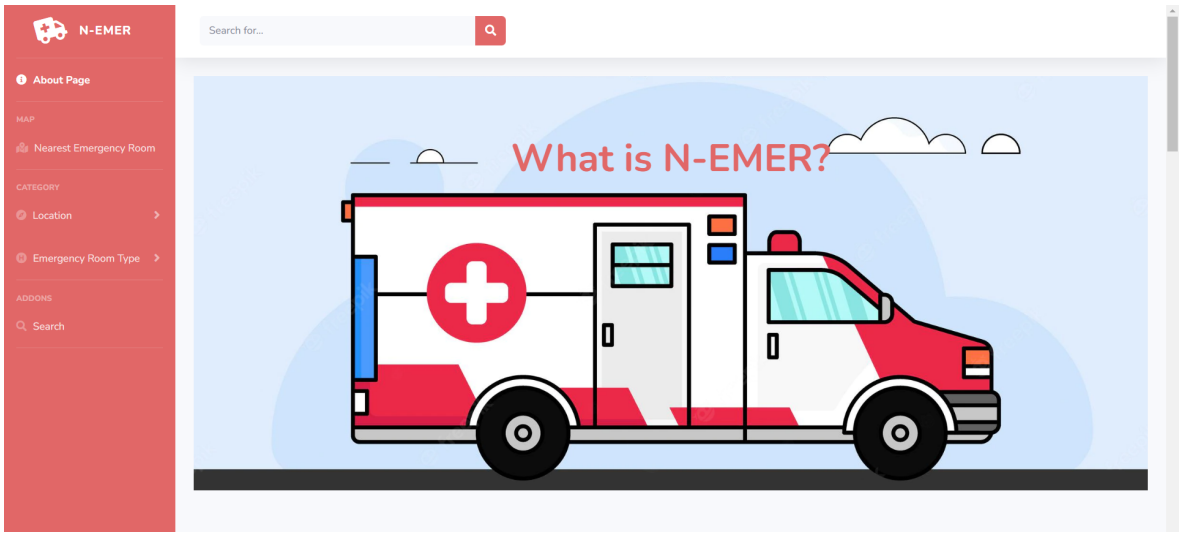
But compared to this population, there are a few medical facilities available in times of emergency. There are only 69 throughout Gyeonggi-do. It is important to locate them because missing the golden time can lead to a number of dangerous situations. Therefore, we came to plan a web project with these capabilities.

We created it using a public data site that stores various data of Gyeonggi-do and Naver Map API.

The current location was measured using the GPS of the device, and the longitude and latitude were setted out by processing json from the emergency medical center. Distance was measured through both longitude and latitude, and the nearest emergency facility could be addressed.

Project Screenshots & Descriptions

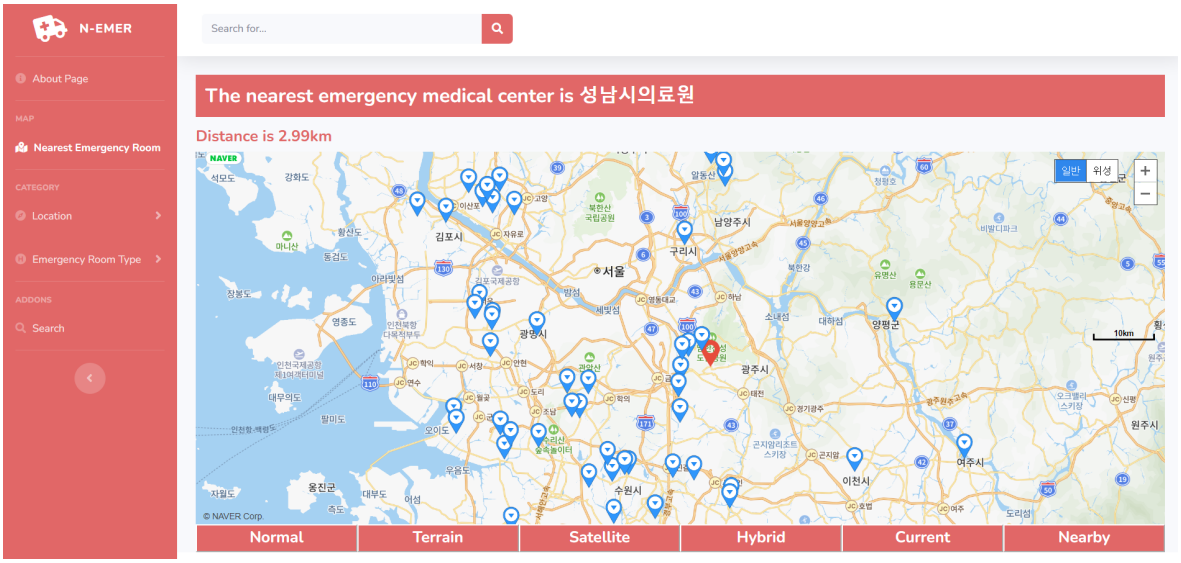
> Main Page



First, the overall structure of the page consists of the search bar at the top of the sidebar on the left, and the content below it. The sidebar can be reduced in size through buttons and can be moved to that page by clicking. Also, click on the logo at the top of the sidebar to navigate to the map page, which is the most important page. Finally, users can enter any keyword such as name, address, and phone number in the search box to find a hospital that fits that keyword.
Now let's look at the contents of the main page with the next image.

The image above is below the main page and contains the purpose, function, and goal of creating the website.

> Map Page



The following is the most important page, the Map page. This is because the page allows emergency patients to find the nearest hospital from their location.
If the user looks at the top of the page, it initially tells the user where the nearest hospital is and then tells the distance between the hospital and the user. Next, when the user looks at the map, the blue marker indicates the emergency room and the red marker indicates the user's location. Users can also click on the blue marker to see the name, address, and phone number of the emergency room. Finally, the first four of the six buttons at the bottom change the way the map is displayed. The fifth button, Current, moves the map to the current user's location. The sixth button, Nearby, moves the map from the current user's location to the nearest hospital.

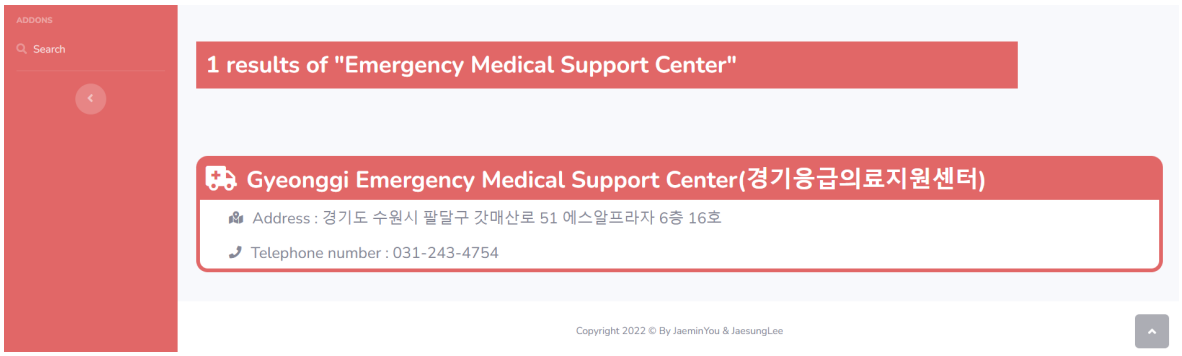
> Category - Location - North Page

The location category is divided into North and South parts, which represent the emergency room located in northern Gyeonggi Province and the emergency room located in southern Gyeonggi Province, respectively. Since the two pages are similar, let's look at the north page.
The picture above is a brief description of the northern part of Gyeonggi-do.

This picture is at the bottom of the north page. First, the user can look at the top to see how many hospitals have been classified as northern. Below that, the hospital's English name, Korean name, address, and phone number are written in card format. Users can also click on the card to enter the hospital's site.
Finally let’s look at the picture, there is an arrow in the lower right corner, which user can press to move to the top of the page.

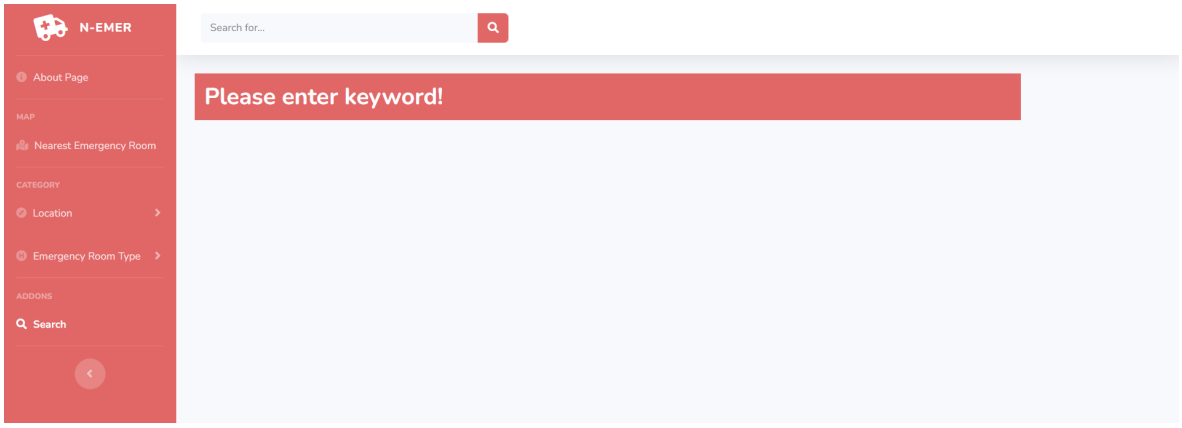
> Category - Emergency Room Type - Medical Support Center

Next, there are four categories in the Emergency Room Type category. Each represents a type of emergency room. Let's look at the Emergency Medical Support Center page among the four.
The format is not much different from the location category. The user can equally view the description of the category and the number of emergency rooms in that category.

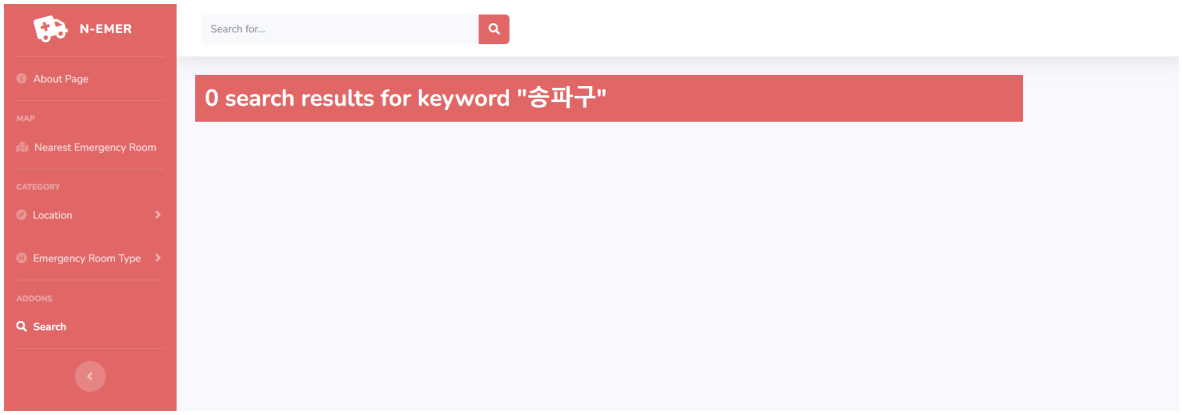


This part is also the same as the location category.

> Search Page



Next is the last page, the Search page. The search bar exists on any page, so if a user presses the search button on any page, the user can go to that page and check the results. The above case is the screen when user first entered the search page.



The picture above is what the search results look like when they don't exist.

Finally, if the input is appropriate, the user will see the number of search results

and the corresponding emergency room in card format.

It was created by referring to the Naver Maps Guide (https://navermaps.github.io/).

Project Implementation

> Naver Map API

We registered with the Web Dynamic Map service of Naver Cloud Platform and obtained the Client ID key. The default ip address for atom-live-server, http://127.0.0.1:3000/, was then added to the list of valid addresses so that you can work locally. And I added the script inside the html. The important thing here is that the port number is fixed at 3000. If you do not start with a fixed port number, the port will change randomly and will not work.

Therefore, if a specific address is not separately entered in the development console in the local environment, Map API authentication will not work.

https://guide.ncloud-docs.com/docs/naveropenapiv3-maps-overview

> Gyeonggi-do public data

We received an authentication key from https://data.gg.go.kr/ (Gyeonggi Data Dream), and got the key value and Json type to the OpenAPI address of Gyeonggi-do emergency medical institutions and emergency medical support centers. The Json has a complicated structure, so we parsed the JsonArray within the JsonArray. We were able to get the name and location of the hospital by parsing the final Json List one more time.

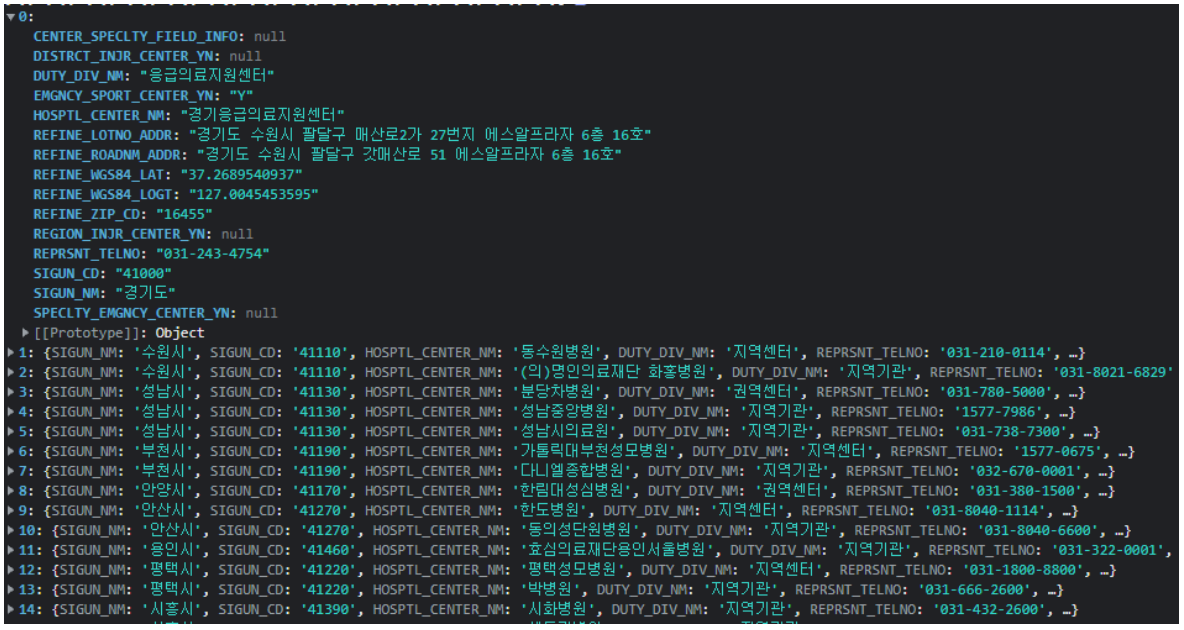
응급의료기관 및 응급의료지원센터 현황

경기도 내 응급의료기관 및 응급의료지원센터 현황입니다.

총평점 ★★★★★ ★★★★★ OK

분류 체계	가족복지보건>보건	데이터 개방일	2018-08-30
태그	경기도 병원, 경기도병원, 경기도 응급실, 경기도응급실, 응급실, 응급의료기관, 응급의료, 응급, 의료기관, 응급의료지원센터, 지원센터, 의료기관	최종 수정일자	2022-03-04
제공 기관	경기도	데이터 기준일자	2022-03-04
제공 부서	보건의료과	갱신 주기	연간
이용 허락 조건	상업적이용허용 및 콘텐츠변경허용	원본 시스템	파일시스템
내용	경기도 내 응급의료기관 및 응급의료지원센터 현황 입니다.		

(json parsing result)




```
▼0:
   CENTER_SPECLTY_FIELD_INFO: null
   DISTRCT_INJR_CENTER_YN: null
   DUTY_DIV_NM: "응급의료지원센터"
   EMGNCY_SPORT_CENTER_YN: "Y"
   HOSPTL_CENTER_NM: "경기응급의료지원센터"
   REFINE_LOTNO_ADDR: "경기도 수원시 팔달구 매산로2가 27번지 에스알프라자 6층 16호"
   REFINE_ROADNM_ADDR: "경기도 수원시 팔달구 갓매산로 51 에스알프라자 6층 16호"
   REFINE_WGS84_LAT: "37.2689540937"
   REFINE_WGS84_LOGT: "127.0045453595"
   REFINE_ZIP_CD: "16455"
   REGION_INJR_CENTER_YN: null
   REPRSNT_TELNO: "031-243-4754"
   SIGUN_CD: "41000"
   SIGUN_NM: "경기도"
   SPECLTY_EMGNCY_CENTER_YN: null
  ▶[[Prototype]]: Object
▶1: {SIGUN_NM: '수원시', SIGUN_CD: '41110', HOSPTL_CENTER_NM: '동수원병원', DUTY_DIV_NM: '지역센터', REPRSNT_TELNO: '031-210-0114', …}
▶2: {SIGUN_NM: '수원시', SIGUN_CD: '41110', HOSPTL_CENTER_NM: '(의)명인의료재단 화홍병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-8021-6829'
▶3: {SIGUN_NM: '성남시', SIGUN_CD: '41130', HOSPTL_CENTER_NM: '분당차병원', DUTY_DIV_NM: '권역센터', REPRSNT_TELNO: '031-780-5000', …}
▶4: {SIGUN_NM: '성남시', SIGUN_CD: '41130', HOSPTL_CENTER_NM: '성남중앙병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '1577-7986', …}
▶5: {SIGUN_NM: '성남시', SIGUN_CD: '41130', HOSPTL_CENTER_NM: '성남시의료원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-738-7300', …}
▶6: {SIGUN_NM: '부천시', SIGUN_CD: '41190', HOSPTL_CENTER_NM: '가톨릭대부천성모병원', DUTY_DIV_NM: '지역센터', REPRSNT_TELNO: '1577-0675', …}
▶7: {SIGUN_NM: '부천시', SIGUN_CD: '41190', HOSPTL_CENTER_NM: '다니엘종합병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '032-670-0001', …}
▶8: {SIGUN_NM: '안양시', SIGUN_CD: '41170', HOSPTL_CENTER_NM: '한림대성심병원', DUTY_DIV_NM: '권역센터', REPRSNT_TELNO: '031-380-1500', …}
▶9: {SIGUN_NM: '안산시', SIGUN_CD: '41270', HOSPTL_CENTER_NM: '한도병원', DUTY_DIV_NM: '지역센터', REPRSNT_TELNO: '031-8040-1114', …}
▶10: {SIGUN_NM: '안산시', SIGUN_CD: '41270', HOSPTL_CENTER_NM: '동의성단원병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-8040-6600', …}
▶11: {SIGUN_NM: '용인시', SIGUN_CD: '41460', HOSPTL_CENTER_NM: '효심의료재단용인서울병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-322-0001',
▶12: {SIGUN_NM: '평택시', SIGUN_CD: '41220', HOSPTL_CENTER_NM: '평택성모병원', DUTY_DIV_NM: '지역센터', REPRSNT_TELNO: '031-1800-8800', …}
▶13: {SIGUN_NM: '평택시', SIGUN_CD: '41220', HOSPTL_CENTER_NM: '박병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-666-2600', …}
▶14: {SIGUN_NM: '시흥시', SIGUN_CD: '41390', HOSPTL_CENTER_NM: '시화병원', DUTY_DIV_NM: '지역기관', REPRSNT_TELNO: '031-432-2600', …}
```


> Http communication (Restful API)

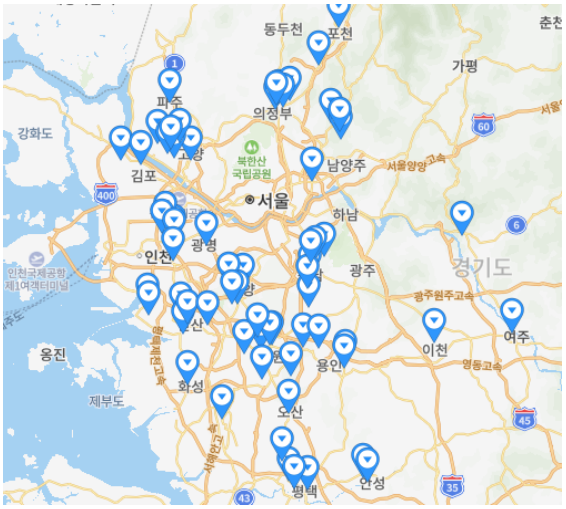
We used xhttp for http communication. Enter the issued authentication key into key, and receive a response in json format. And after receiving it, we parsed it and got the information of a total of 69 centers.

```
xhttp.open("GET", "https://openapi.gg.go.kr/EmgncyMedcareInstStus?Key=a016e5b03
xhttp.send();
```

```
var obj=JSON.parse(this.responseText);
hospi_list = obj.EmgncyMedcareInstStus[1].row;
```

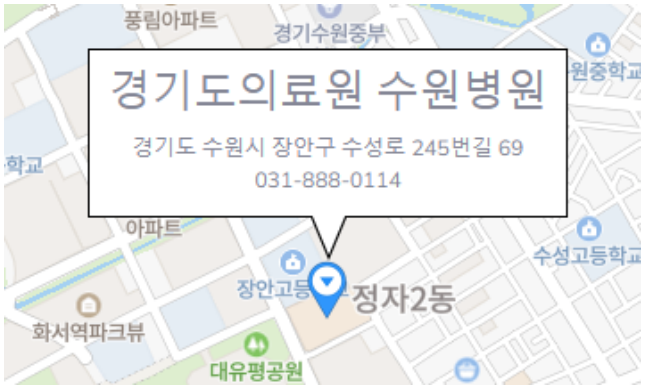
Then we created a marker object, put it in a global array, and pushed only the information we needed. At this time, information and markers were stored separately.

```
for(var i =0;i < hospi_list.length;i++){
    var marker = new naver.maps.Marker({
        position: new naver.maps.LatLng(hospi_list[i]
        map: map,
        title: hospi_list[i].HOSPTL_CENTER_NM
    });

    var infoWindow = new naver.maps.InfoWindow({
      // marker infomation part
        content: '<div style="width:auto;text-align:c
        +'</h4>'+ hospi_list[i].REFINE_ROADNM_ADDR +
        + hospi_list[i].REPRSNT_TELNO + '</div>',
        borderColor: "#000000",
        borderWidth: 1


    });

    markers.push(marker);
    infoWindows.push(infoWindow);
    korName.push(hospi_list[i].HOSPTL_CENTER_NM);

}
```



Finally, we attached a click handler to the marker to make the information window work.

```
for (var i=0, ii=markers.length; i<ii; i++) {
    naver.maps.Event.addListener(markers[i], 'click', getClickHandler(i));
}
```



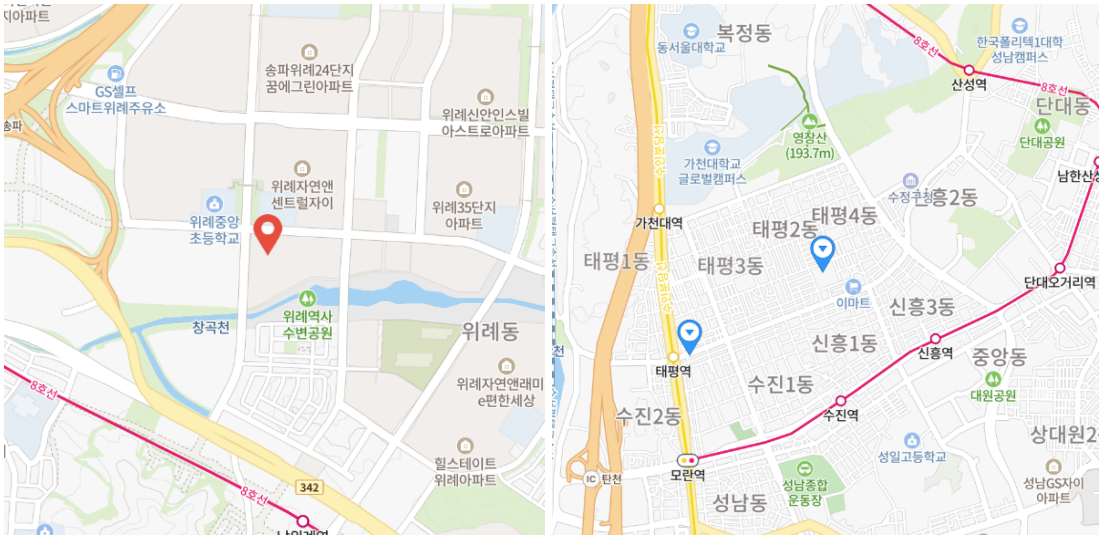
<-closest to SKKU

In addition, buttons providing six convenient functions were added under the map.

Normal	Terrain	Satellite	Hybrid	Current	Nearby

We can change the map type through the Normal, Terrain, Satellite, and Hybrid buttons. If we press the Current button, the map moves to the current location, and the location of the map changes to the location of the nearest emergency center in Nearby.

It is the screen of Normal, Terrain, Satellite, and Hybrid in order.



The current location is indicated by a red marker on the map. If you press the Current button, you can return to the location where you were. Also, if you press the Nearby button, the nearest hospital is displayed in the center of the screen.

In addition, it automatically measures the name and distance of the nearest hospital from the current location as follows when location permission is initially granted.

```
$(window).on("load", function() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(onSuccessGeolocation, onErrorGeolocation);
    } else {
        var center = map.getCenter();
        infowindow.setContent('<div style="padding:20px;"><h5 style="margin-bottom:5px;colo
        infowindow.open(map, center);
    }

});
```

After loading the latitude/longitude of the saved hospital list, the distance was calculated using the latitude/longitude of the current location and the following formula. So we were able to find the nearest hospital.

$$Point_1 = [lat_1,\ lon_1];$$

$$Point_2 = [lat_2,\ lon_2];$$

$$\delta_{lat} = radian(lat_1) - radian(lat_2);$$

$$\delta_{lon} = radian(lon_1) - radian(lon_2);$$

$$\varphi = 2 \times \arcsin\left(\sqrt{\sin^2\left(\frac{\delta_{lat}}{2}\right) + \cos(radian(lat_1)) \times \cos(radian(lat_2)) \times \sin^2\left(\frac{\delta_{lon}}{2}\right)}\right);$$

$$Distance = \varphi \times R;\ R = Average\ Earth\ Radius$$

```
var location = new naver.maps.LatLng(position.coords.latitude,
                                     position.coords.longitude);
```

```
curLat = position.coords.latitude;
curLon = position.coords.longitude;

var meter = 100000;
var min = 0;
for(var i = 0; i < hospi_list.length ;i++){
  var near = getDistance(curLat, curLon, hospi_list[i].REFINE_WGS84_LAT, hospi_list[i].REFINE_WGS84_LOGT);
  if(meter > near){
      min = i;
      meter = near;
  }
}
```

```
Nearest EM print
console.log(hospi_list[min].HOSPTL_CENTER_NM);
var nearEM = document.getElementById("nearEM");
nearEM.innerText = "The nearest emergency medical center is "+hospi_list[min].HOSPTL_CENTER_NM;
var disEM = document.getElementById("disEM");
disEM.innerText = "Distance is "+String(meter.toFixed(2)) + "km";

nearLat = hospi_list[min].REFINE_WGS84_LAT;
nearLon = hospi_list[min].REFINE_WGS84_LOGT;
```

The nearest emergency medical center is 성남시의료원

Distance is 2.99km

Impression

We wanted to create a meaningful result by doing a project to finish this semester. Also, we wanted it to be usable in real life and to have the purpose of pursuing the public good.

We made the design easy to understand at a glance, and placed elements with functions to make it easy to operate. The most difficult part of the implementation process was the map setting part. Unlike the planning, there were parts that did not work well in small parts.

It was especially tricky when dealing with the marker part. First, the longitude and latitude of the emergency center were required to display the marker at the desired location.

Fortunately, the public data API had that data. However, after sending the post to the API, the response Json file was composed of multiple Json Lists, so it was difficult to process. The internal Json List was also made up of Json Lists. Machining allowed us to obtain the coordinates of each center. Here, we needed to mark a number of markers on the map with these longitudes and latitudes. It was quite cumbersome to create markers through that data. By simply calling the function repeatedly, the marker was not created and a crash occurred, and even if

it was created, it was not saved. We were thinking of putting each one as many as I could. Since it was not possible to create dozens of markers one by one, we had to find a way, and in the end, We were able to implement the function by repeatedly creating marker objects, inserting information, and then storing the markers in an Array.

Also, the function to display the current location did not originally exist, and it did not work if configured with the same method as the existing marker. So, We wrote a separate overlay code and implemented it by overlapping it on the map.

Lastly, since we work on a project by dividing roles with team members, it seems to be more efficient because I can take responsibility for each other's part and pour energy into it. Even if a defect occurs, it can be handled in a decentralized manner, which is very convenient for trouble management.