

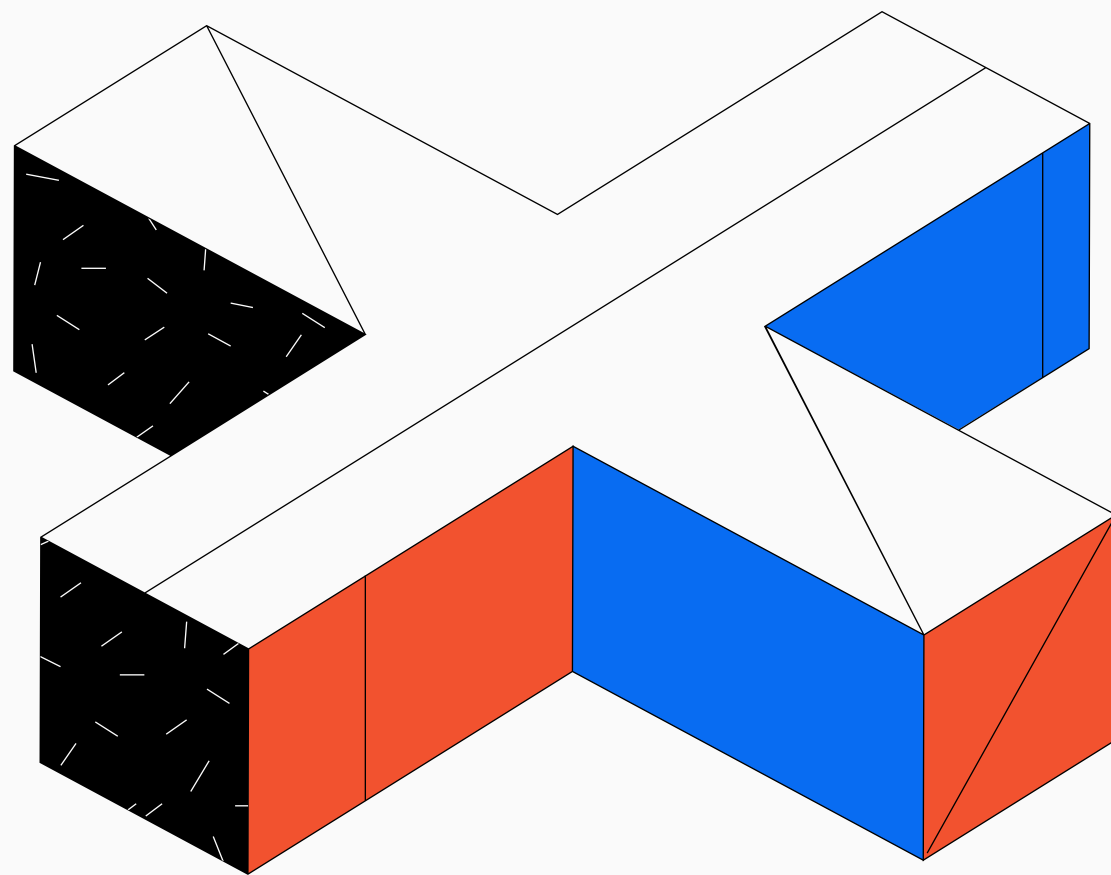
병원 관리 시스템

Hospital Management



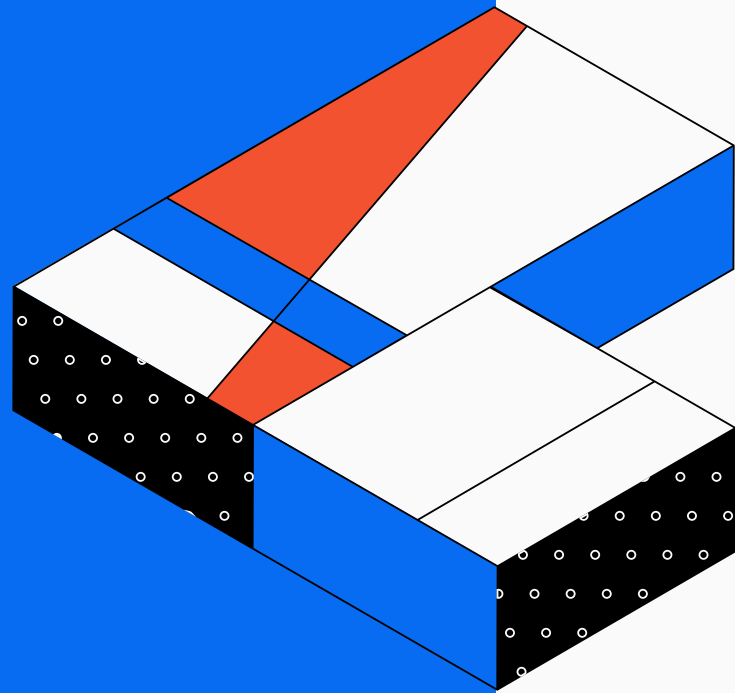
박경한 이예찬 김종현
오병현 차민주

목차

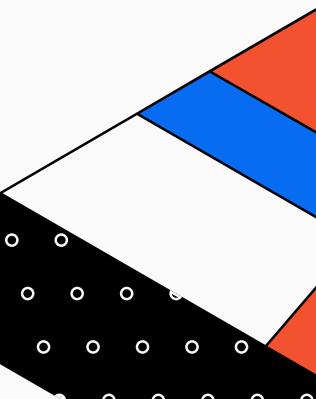


- 팀 목표
- 팀 로드맵
- 구현 화면
- 코드 리뷰
- 자체 피드백
- 마침

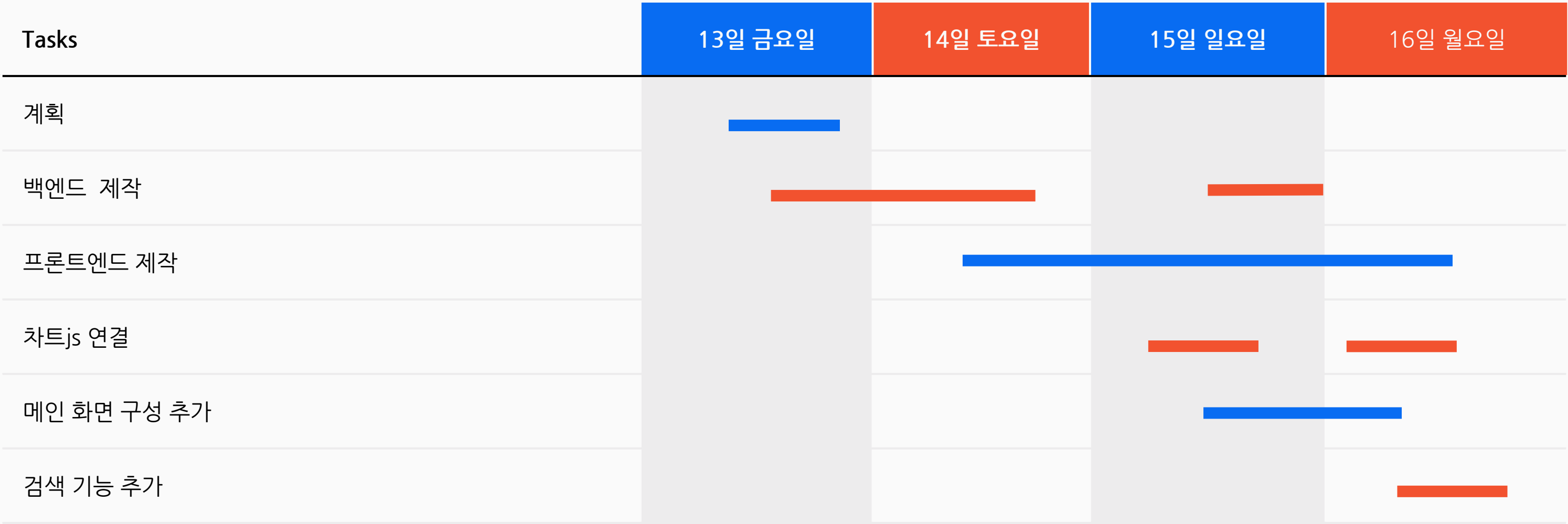
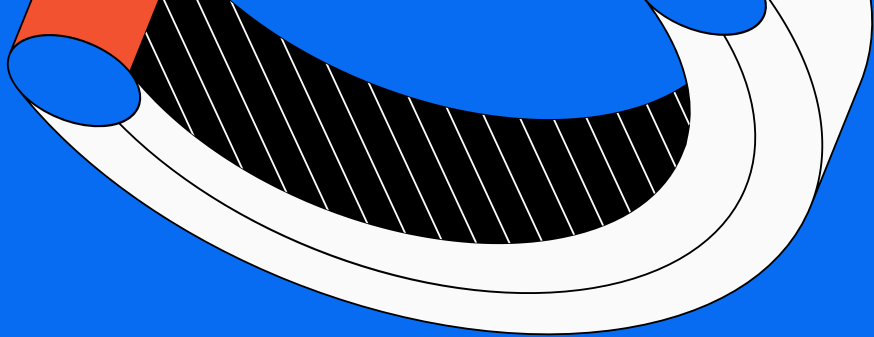
팀 목표

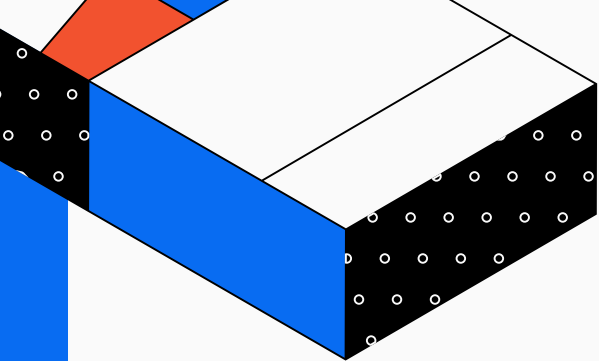


- **목표**
심플하면서 차트js를 통한 직관적인 UI구성
- **목표 2**
등록과 검색 연결로 빠른 탐색
- **목표 3**
외부 라이브러리 (부트스트랩) 적극 활용으로 디자인 시간 단축

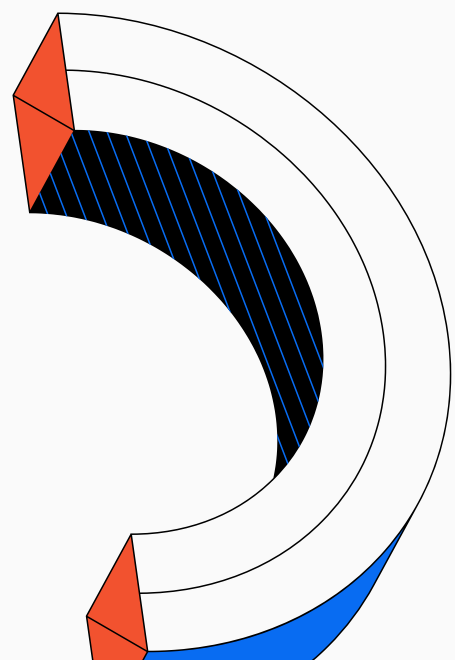
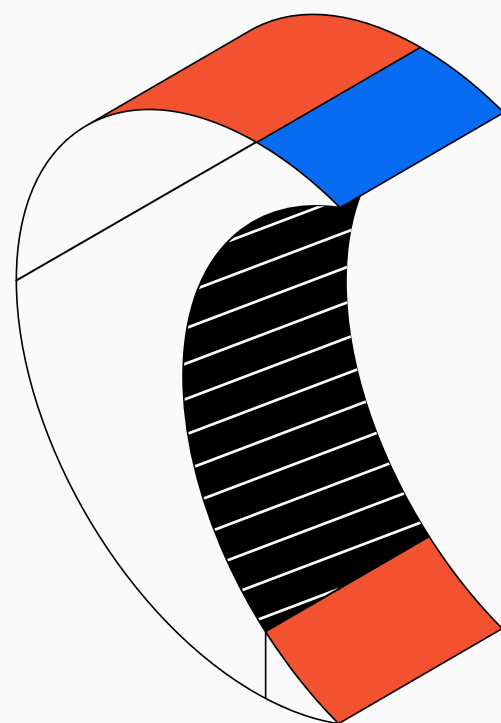


팀 로드맵

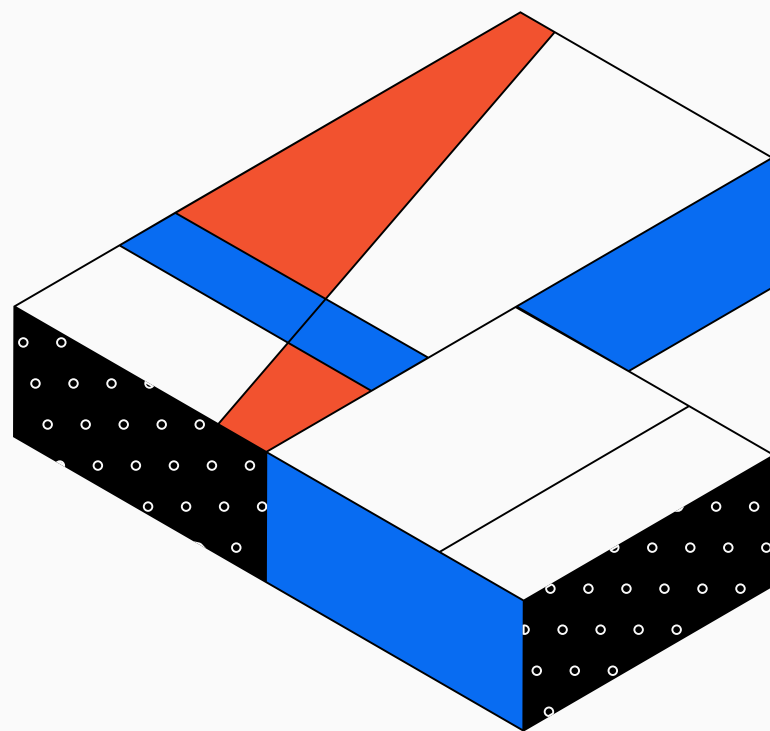




웹 시연



코드 리뷰



코드 리뷰

```
public class Facility extends General {  
    · private String company;          // 회사 이름  
    · private String installedPlace;   // 설치 장소  
    · private int condition;           // 정상 작동 여부  
    · private int price;  
    · private int stock;               // 재고 수량  
    · private int checkCycle;          // 점검 사이클  
}
```

```
public class Board extends General {  
    · private String title;  
    · private String contents;  
}
```

```
public class Patient extends General {  
    · private String symptome; // 증상  
    · private int bloodType; // 혈액형  
    · private float bodyHeat; // 체온  
    · private int bloodPressure; // 혈압  
    · private int height; // 키  
    · private int weight; // 몸무게  
    · private boolean oversea; // 최근 14일간 해외 여행 여부  
    · private boolean insurance;  
    · private boolean hospitalized;  
}
```

```
public class Staff extends General {  
    · private String position;  
    · private int department;  
    · private char holiday;  
    · private int annualPay;  
}
```

코드 리뷰

```
public abstract class General {  
    private int no;  
    private String name;  
    private String tel;  
    private String basicAddress;  
    private String detailAddress;  
    private int birthDate;  
    private char gender;  
    private String createDate;  
}
```

Generalization

유지보수를 쉽게 하기 위해
환자, 직원, 기기
클래스 들이 상속받도록
추상 클래스 사용

코드 리뷰

총 환자 수

2명



입원 환자 수

2명

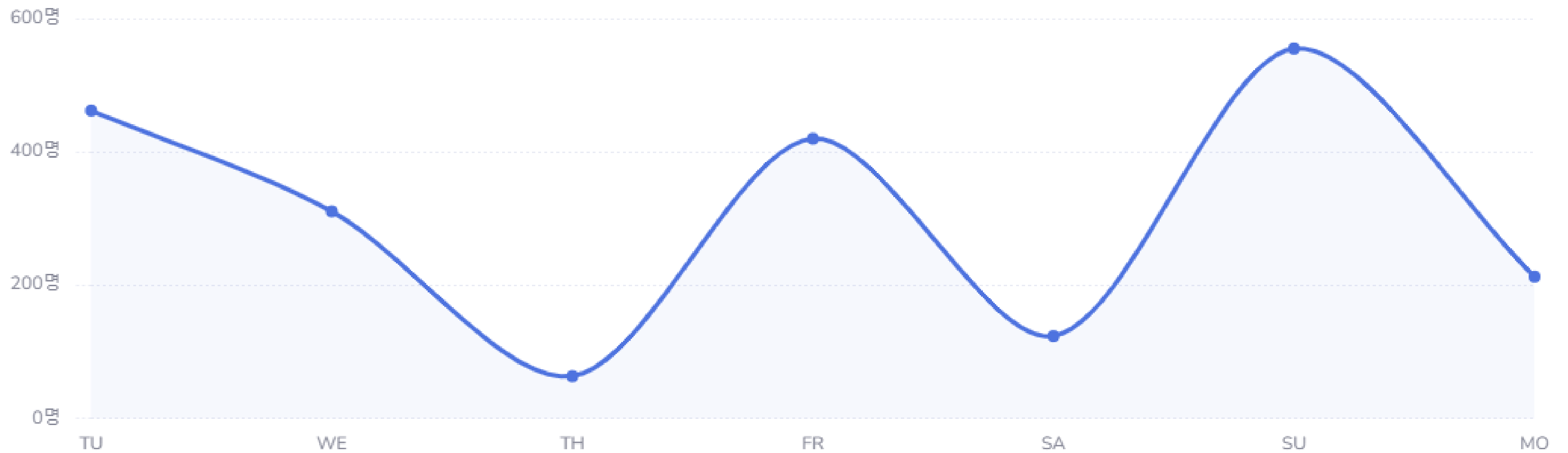


```
fetch(`http://localhost:8080/patients`)
  .then((response) => {
    return response.json();
  })
  .then((obj) => {
    var cnt = 0;
    for (var i of obj.data) {
      if (i.hospitalized == true) {
        cnt++;
      }
    }
    document.querySelector('#hoscnt').innerHTML = `${cnt}명`;
  })
  .catch((err) => {
    alert('서버 요청 오류!');
    console.log(err);
  });
```

```
fetch('http://localhost:8080/patientCnt')
  .then((response) => {
    return response.json();
  })
  .then((obj) => {
    var html = `${obj.data}명`;
    var cnt = document.querySelector('#patientcnt');
    cnt.innerHTML = html;
  })
  .catch((err) => {
    alert('서버 요청 오류!');
    console.log(err);
  });
```

코드 리뷰

요일별 환자 수



코드 리뷰

```
fetch(`http://localhost:8080/patients`)
  .then((response) => {
    return response.json();
  })
  .then((obj) => {

    var week = ['SU', 'MO', 'TU', 'WE', 'TH', 'FR', 'SA'];
    var su = 0; var mo = 0; var tu = 0; var we = 0; var th = 0; var fr = 0; var sa = 0;
    for (var i of obj.data) {
      var dayOfWeek = week[new Date(i.createdDate).getDay()];
      console.log(dayOfWeek);
      if (dayOfWeek == 'SU') {
        su++;
      } else if (dayOfWeek == 'MO') {
        mo++;
      } else if (dayOfWeek == 'TU') {
        tu++;
      } else if (dayOfWeek == 'WE') {
        we++;
      } else if (dayOfWeek == 'TH') {
        th++;
      } else if (dayOfWeek == 'FR') {
        fr++;
      } else if (dayOfWeek == 'SA') {
        sa++;
      }
    }
  })
}
```

```
var today = week[new Date().getDay()];
console.log(today);
```

```
var yoilarr = ['MO', 'TU', 'WE', 'TH', 'FR', 'SA', 'SU'];
var cntarr = [mo, tu, we, th, fr, sa, su];
```

```
for (var i = 0; i < 7; i++) {
  yoilarr.push(yoilarr[0]);
  yoilarr.shift();
  cntarr.push(cntarr[0]);
  cntarr.shift();
  if (today == yoilarr[6]) {
    break;
  }
}
```

코드 리뷰

장비 상태 현황



● 매우 양호 ● 양호 ● 점검필요 ● 불량

```
fetch(`http://localhost:8080/facilities`)
  .then((response) => {
    return response.json();
  })
  .then((obj) => {
    var cnt1 = 0;
    var cnt2 = 0;
    var cnt3 = 0;
    var cnt4 = 0;
    for (var i of obj.data) {
      if (i.condition == 1) {
        cnt1++;
      } else if (i.condition == 2) {
        cnt2++;
      } else if (i.condition == 3) {
        cnt3++;
      } else if (i.condition == 4) {
        cnt4++;
      }
    }
    var ctx = document.getElementById('myPieChart');
    var myPieChart = new Chart(ctx, {
      type: 'doughnut',
      data: {
        labels: ['매우 양호', '양호', '점검필요', '불량'],
        datasets: [
          {
            data: [cnt1, cnt2, cnt3, cnt4],
```

자체 피드백

성과

- 메인 화면 차트 구성을 통해 몰랐던 지식을 알게 되었고
- 계획 했던 화면 구성이라 괜찮은것 같다

주제 한계

- 처음에는 아이디어가 제한적이라 환자 직원 등록 관리 기능만 해보았는데 만들면서 보니까 병원에서 관리하는 기능들이 생각보다 많았다

차트 부실

- 데이터베이스 없이 자체서버로 인해 차트를 볼륨감 있게 보기 어려움 추후 sql을 배운다면 db연동 계획

로그인 기능 상실

- 관리시스템이라 관리자만 만질거라 생각해 기능을 배제해버린 요인



감사합니다

