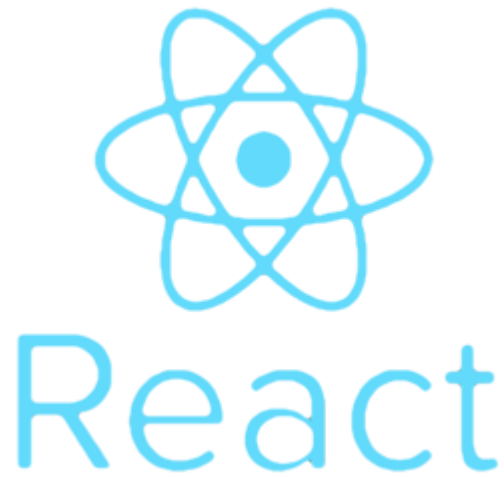


# 풀스택 경험해보기



# 목차

## 1. 프로젝트 Cloud

- 목표
- 프로젝트 구조
- 기획안
- 문제점
- 인증, 파일 매니저, 채팅
- 시연
- 피드백

## 2. 프로젝트 GoodChoice

- 요구사항
- 프로젝트 구조
- 구현 조건
- 데이터 모델
- 기획안
- 달력 미션
- 필터링 미션
- 시연

# Project Cloud - 목표

1. 백엔드 구축 :Node.js + Socket.io + MongoDB

2. 프론트 엔드 구축 : React.js

- Redux / MobX 등 사용하지 않고 상태 관리

3. 회원가입 / 로그인 / 로그아웃 기능

- 세션 생성으로 로그인 유지

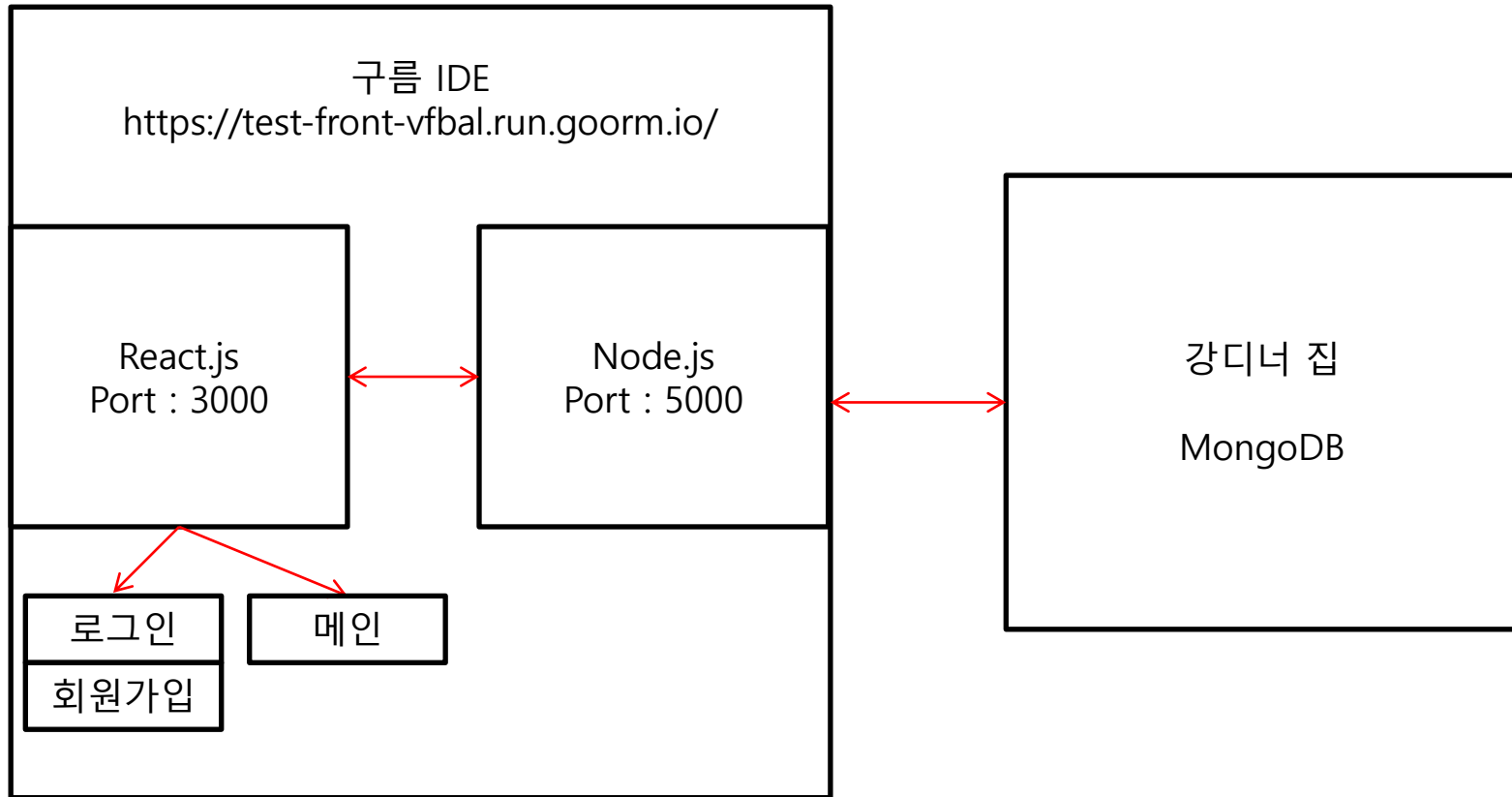
4. 파일 매니저 기능

- zip, tar 지원
- Tree 구조 UI
- 파일 읽기, 쓰기

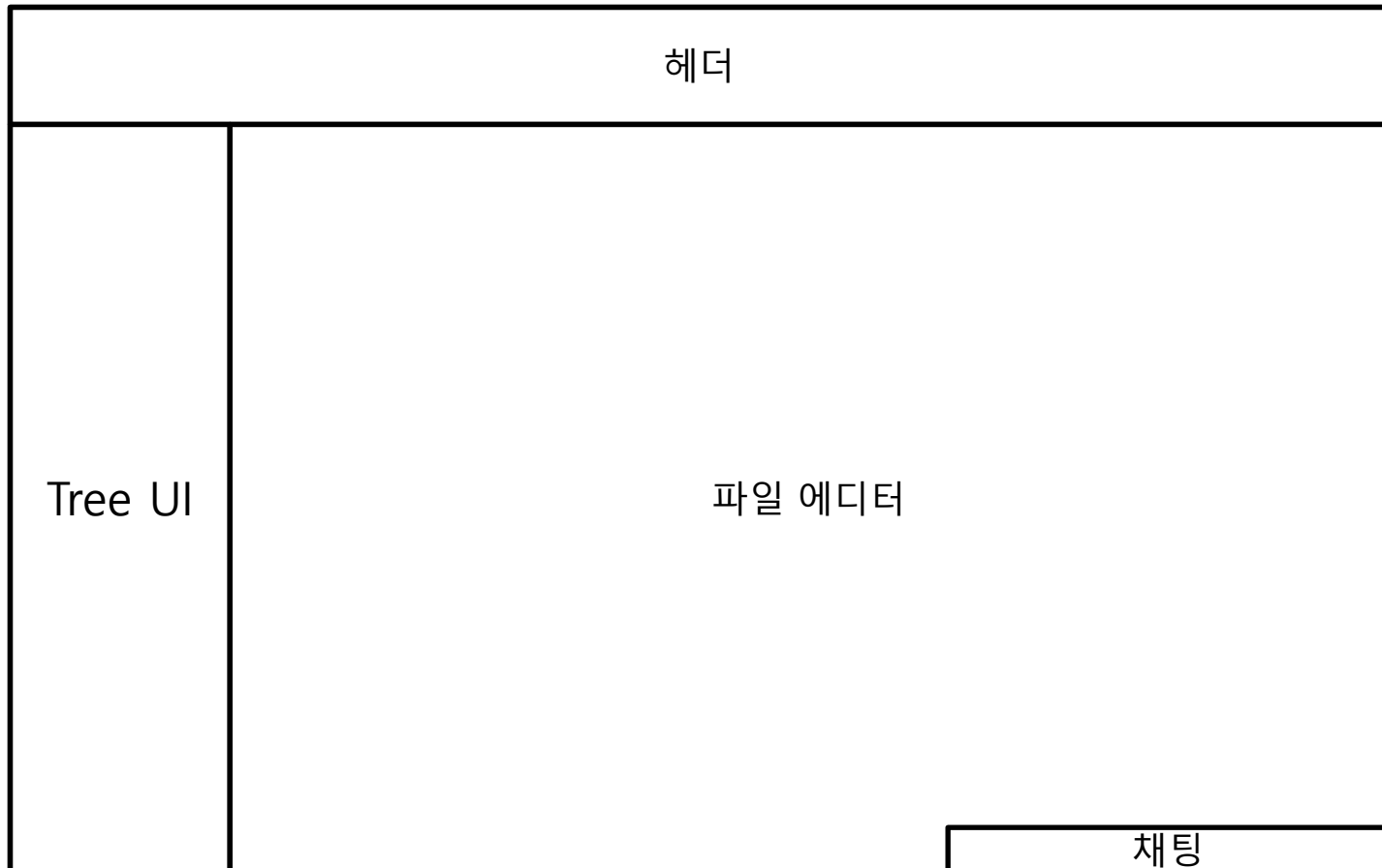
5. 채팅 기능

- 전체 채팅
- 귓속말

# Project Cloud - 프로젝트 구조



# Project GoodChoice - 기획안



# Project Cloud - 문제점 1

**React 처음인데...**

# Project Cloud - 문제점 2

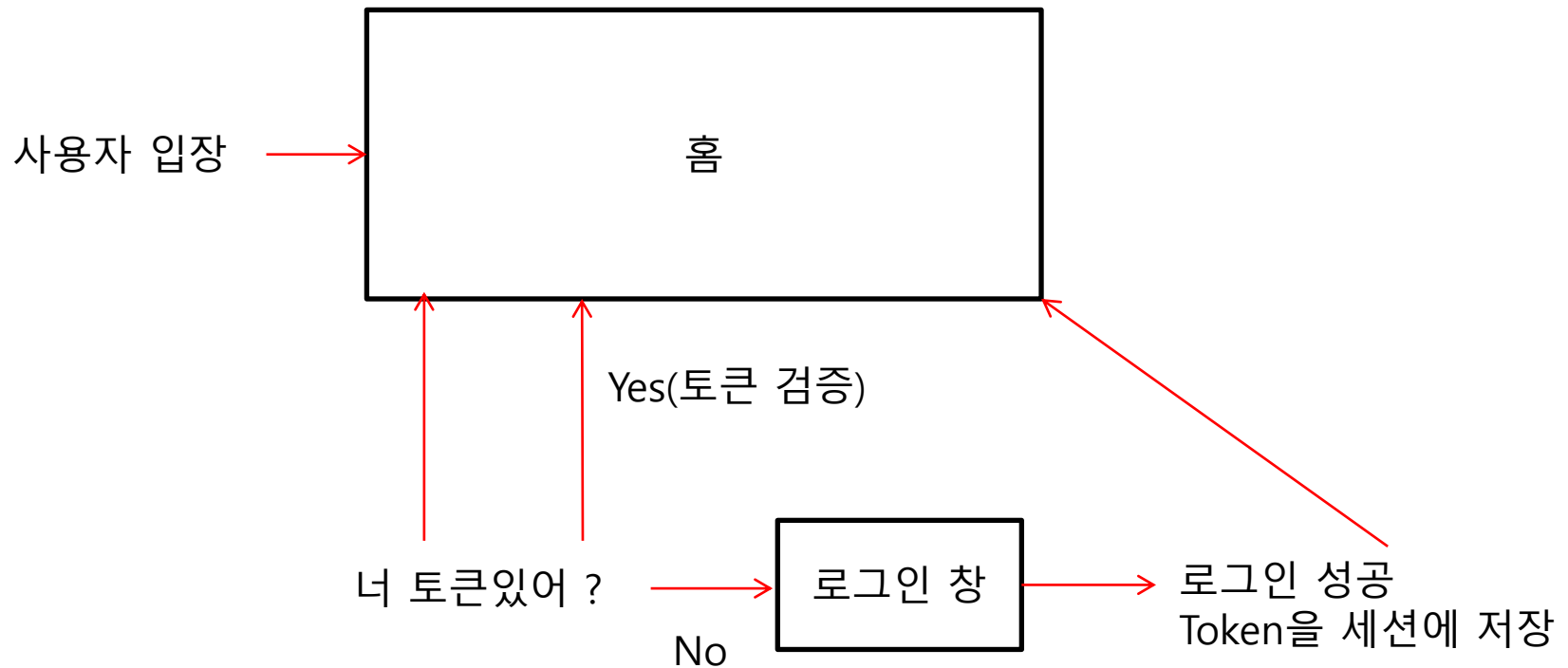
프론트 + 백엔드 **VS**

프론트 따로

백엔드 따로

# Project Cloud - 인증

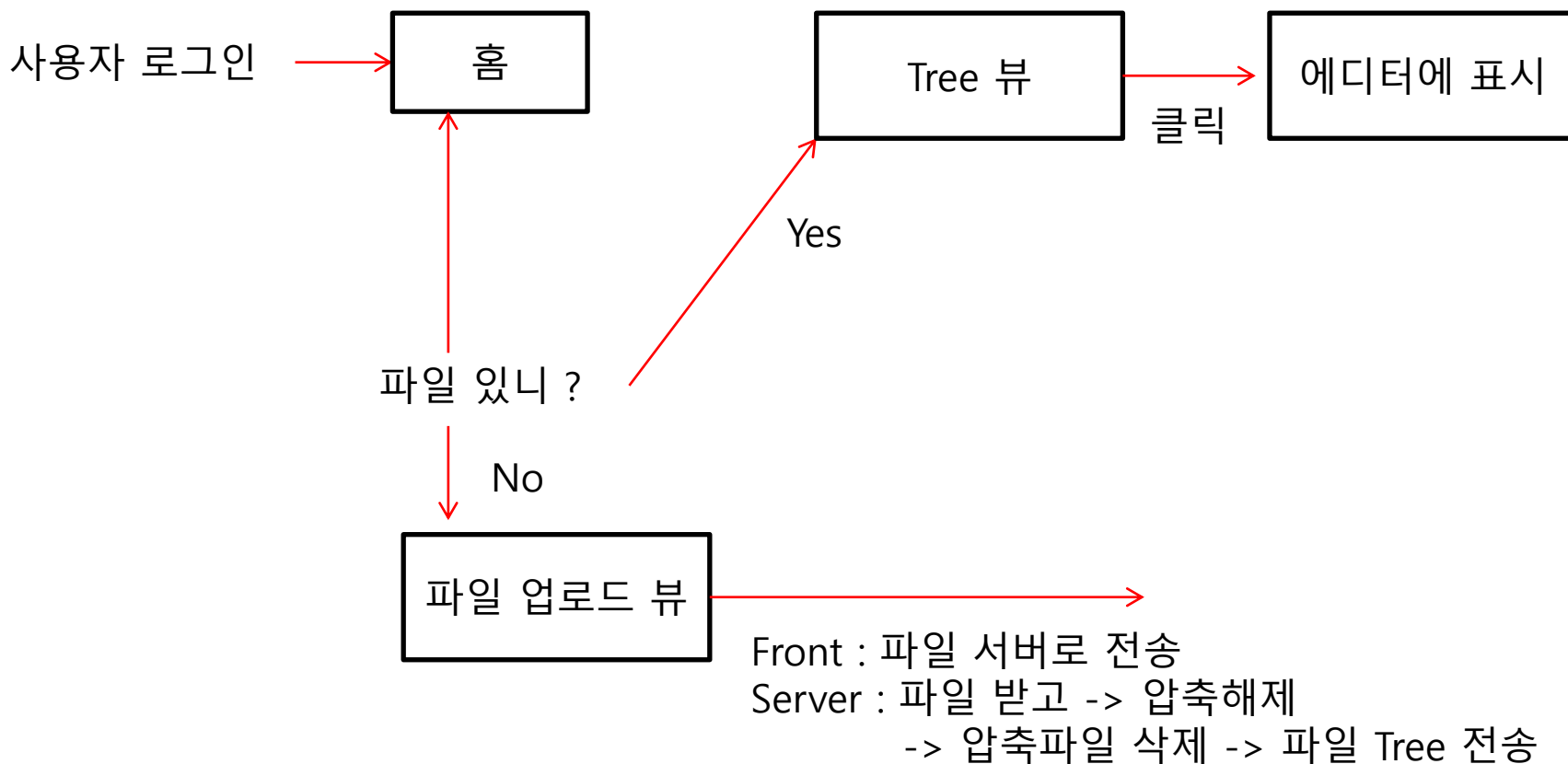
Json Web Token 인증





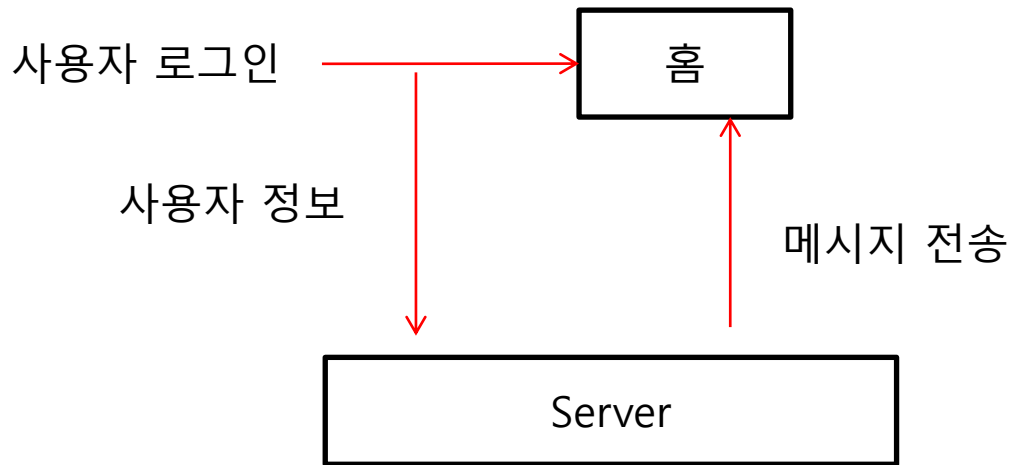
# Project Cloud - 파일 매니저

upload/(ID) 디렉토리에 저장



# Project Cloud - 채팅

Socket.io 이용한 실시간 채팅



# Project Cloud - 시연

<https://test-front-vfbal.run.goorm.io/>

로그인

회원가입

파일 업로드

파일 수정

채팅

# Project Cloud - 피드백

1. 에러 처리 로직이 불안한 경우가 많다.
2. 채팅 기능 외 다른 기능이 제대로 실행 되지 않음.
3. 최신 node 버전의 문법을 적극적으로 사용하지 않음.
4. 불필요한 console.log 가 많음.

# Project GoodChoice - 목표

1. 모든 데이터는 가데이터로 하되, 명시 된 필터 조건은 모두 구현되어야 함

→ 고민거리

- 프론트에서 데이터 전체를 가지고 필터링을 원하는건가 ?
- 백엔드를 간단하게 구축하고, MongoDB 쿼리를 이용하면 편리할 것 같은데 괜찮을까 ?
- > 둘다 하자 !

2. React, Vue.js, Angular 등 어떤 것으로 구현해도 상관 없으나 사용한 라이브러리의 사용 이유나 프로젝트의 구조 설명이 잘 되어있길 희망

→ Angular

- Angular3 아래 버전과 Angular4+ 버전 차이가 크고, 지원 모듈 정보가 별로 없음.
- 다시 익히기에 시간이 오래 걸림.

Vue

- 매우 빠른 프로토타입
- Template 형태 + JSX 문법 혼용

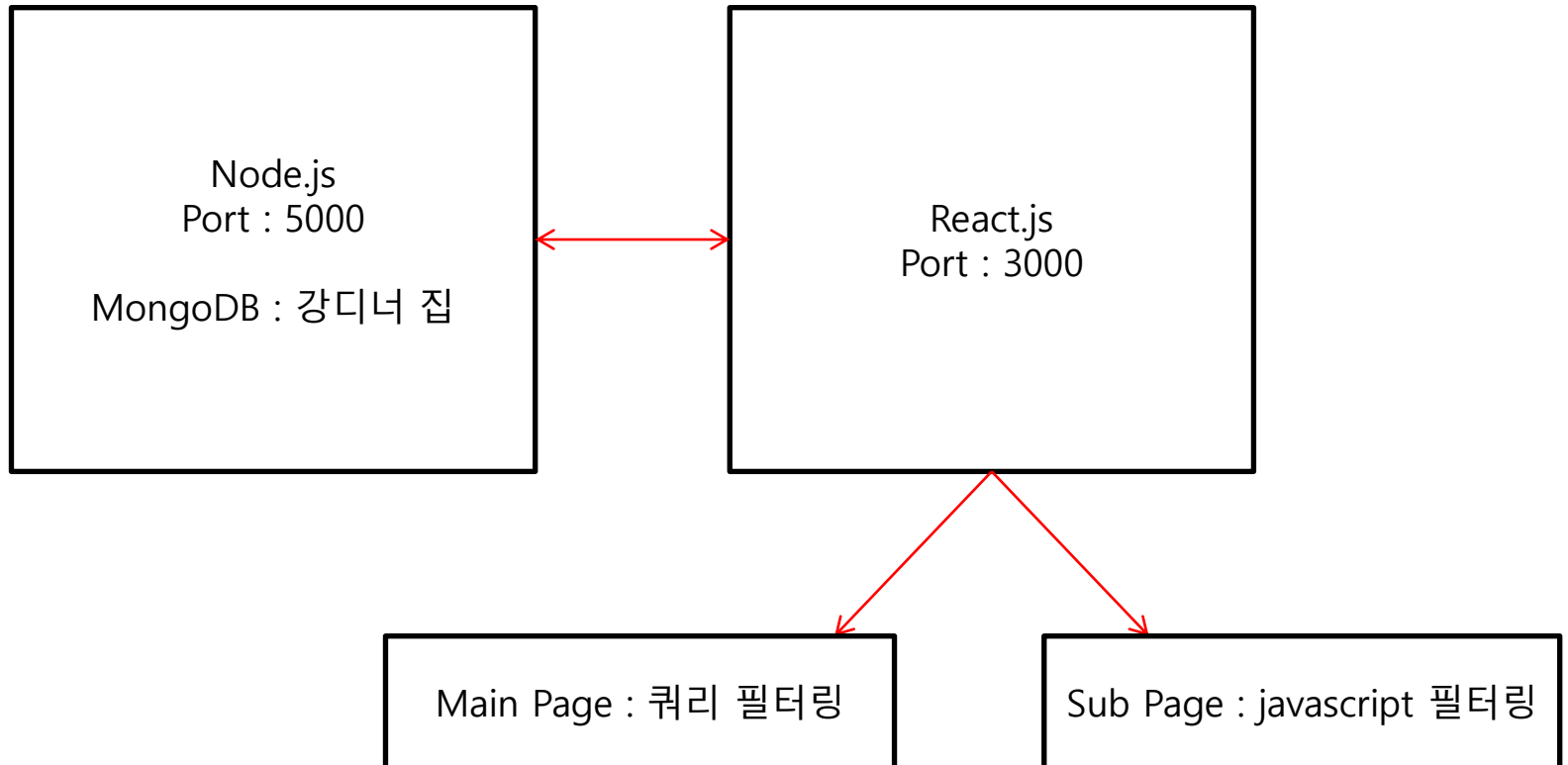
React

- Server + Client 동시 개발 경험
- JSX 문법
- 간단한 상태관리 -> state, setState

# JSX 문법

```
<article className="result_container">
  <ul>
    {this.state.searchResult.map( (c, idx) =>
      <li className="result_list" key={idx}>
        <div className="result_common">이름 : {c.name}</div>
        <div className="result_common">캠핑 유형 : {c.type}</div>
        <div className="result_common">예약 가능 : {c.reservation ? '가능' : '불가능'}</div>
        <div className="result_common">최대 인원 : {c.maxPersonnel}</div>
        <div className="result_common">가격(1박) : {c.price}</div>
        <div className="result_common">숙박 가능 일자 : <span className="show_possibleDate" onClick={this.showPossibleD
          <div className="possibleDate_area" ref={'area_'+idx}>
            {c.possibleDate.map( (date, id) =>
              <div className="result_possibleDate" key={id}>{this.changeResultDate(date)}
            </div>
            </div>
          </div>
        </div>
      </li>
    )}
  </ul>
</div>
```

# Project GoodChoice - 프로젝트 구조



# Project GoodChoice - 구현 조건

날짜

4.17 ~ 4.18 · 1박

상세조건

초기화 적용

예약 가능

캠핑 유형

오토캠핑

글램핑

카라반

인원 2

가격 1만원 이상

1만원 30만원

1. 날짜 (달력 UI)
2. 가격 (범위 지정 UI)
3. 예약 가능 여부
4. 캠핑 유형
5. 인원



# Project GoodChoice - 데이터 모델

1. 날짜 (달력 UI)
2. 가격 (범위 지정 UI)
3. 예약 가능 여부
4. 캠핑 유형
5. 인원

```
[  
  {  
    "name" : "잠실 첼로",  
    "type" : "auto",  
    "reservation" : true,  
    "maxPersonnel" : 4,  
    "possibleDate" :  
      [20190602, 20190603],  
    "price" : 30000  
  }  
]
```

# Project GoodChoice - 기획안

여기 어때 ?				
날짜	가격	예약 가능	캠핑 유형	인원
			초기화	적용
결과				

# Project GoodChoice - 달력 미션

기획의도			
날짜 선택	체크인 선택 시	date.select-1	체크아웃은 리셋한다. (미선택 상태로 변경)
		date.select-2	하단 버튼은 체크인 날짜만 있는 date.confirm-1 상태
		date.select-3	선택한 체크인 이전 날짜 활성화
		date.select-4	체크인일로부터 최소숙박일(m)~최대숙박일(n)+1 구간 활성화 (체크아웃을 선택해야 하기 때문에 n박 다음 날짜까지 활성화 필요)
		date.select-5	체크인 선택 상태에서, 체크인보다 앞선 날짜 선택 시 - 체크인일을 변경한다. - date.select-2 과 동일한 결과
		date.select-6	체크인 선택 상태에서, 체크인보다 뒤의 날짜 선택 시 - 체크아웃 선택 - date.select-3 과 동일한 결과
	체크아웃 선택 시	date.select-7	체크인 날짜 때문에 비활성 시켰던 11일 이후 날짜는 date.active1~7 에서 정의된 최초 진입 시 활성화/비활성 상태로 변경
		date.select-8	하단 버튼 date.confirm-2 상태

## select-1

- **체크인**, **체크아웃**이 선택되어 있으면
- **체크아웃**은 초기화 후 선택지를 **체크인**으로 변경

## select-2

- **체크인** 선택하면 하단 버튼 Text 수정

## select-3

의미를 몰랐지만 -> select-6 보고 확인(**체크 아웃** 선택)

## select-4

- (m) 은 미정, (n)은 confirm-2에 1~9라고 표시
- **체크인** 선택 시 (9)+1 구간 활성화

## select-5

- **체크인** 선택 상태에서 **체크인**보다 앞선 날짜 선택 시
- **체크인** 날짜 변경 후 이전 **체크인**을 **체크아웃**으로 변경

## select-6

- **체크인** 선택 상태에서, **체크인**보다 뒤에 날짜 선택 시
- **체크 아웃** 선택

## select-7

- **체크아웃** 선택 시 비활성화를 활성화로 변경

## select-8

- **체크아웃** 선택 시 하단 버튼을 confirm-2 상태로 변경

# Project GoodChoice - 달력 미션

선택 완료 버튼	기획의도		
	체크인 선택 Y & 체크아웃 선택 N 경우	date.confirm-1	- 버튼 활성화 상태 - 버튼명 표기 : {m월 n일(요일)} 체크인 검색 - 버튼 선택 시 체크아웃일은 최소숙박일에 따라 date.confirm-4~5 와 같이 보고 검색
		date.confirm-4	최소숙박일(m) = 1 일 때 --> 체크아웃일 = 체크인 다음날
		date.confirm-5	최소숙박일(m) > 1 일 때 --> 체크아웃일 = {m}일 뒤 예) 9/1(토) 체크인 선택, 최소숙박일 2일 --> 체크아웃 9/3(월)이상 선택시 버튼이 활성화 됨
	체크인 선택 Y & 체크아웃 선택 Y	date.confirm-2	- 버튼 활성화 상태 - 버튼명 : {월/일(요일)} ~ 월/일(요일) n박 선택 - n : 1~9
	체크인 선택 N & 체크아웃 선택 N	date.confirm-3	- 버튼 비활성화 상태 - 버튼명 : 날짜를 선택해주세요.

## confirm-1

버튼명을 m월 n일(요일) **체크인** 으로 변경  
버튼 클릭 시 이벤트 없음

## confirm-4

최소숙박일을 지정하지 않음.  
0박일 경우 버튼 비활성화

## confirm-5

**체크인** 선택 시 (9)+1 구간 활성화  
버튼 클릭 시 이벤트 없음

## confirm-2

버튼명 : 5월 30일 (목) ~ 6월 2일 (일) / 3박 4일  
버튼 활성화

## confirm-3

버튼 클릭 시 이벤트 없음  
버튼명 : 날짜를 선택해주세요.

# Project GoodChoice - 필터링 미션

javascript 를 이용한 필터링

타입 && 예약 여부 && 최대 인원 && 가격  
+ 숙박 가능 일자

```
rangeValue: { min: 1, max: 30},
reservation : false,
autoCamp: false,
glam: false,
karaban : false,
option_people: 1,
duringDay: [],
```

```
let type= [];
if(this.state.autoCamp) type.push('auto');
if(this.state.glam) type.push('glam');
if(this.state.karaban) type.push('karaban');

if(!this.state.fixedDay) return alert('입실 날짜와 퇴실 날짜를 입력해주세요.');
```

if(type.length ===0) return alert('캠핑 타입을 선택해주세요.');

const data = this.state.data; —→ 임의로 만든 가데이터  
let searchResult = [];

```
for(let i=0, len=data.length; i<len; i++){
  if(type.indexOf(data[i].type) > -1 && data[i].reservation === reservation
    && data[i].maxPersonnel <= people && data[i].price <= maxPrice*10000 && data[i].price >= minPrice*10000
  ){
    let dayFilter = intersect(data[i].possibleDate, duringDay);
    if(JSON.stringify(dayFilter) === JSON.stringify(duringDay)) searchResult.push(data[i]);
  }
}
```

// 교집합 함수

```
function intersect(a, b) {
  var tmp={}, res=[];
  for(var i=0; i<a.length; i++) tmp[a[i]]=1;
  for(var i=0; i<b.length; i++) if(tmp[b[i]]) res.push(b[i]);
  return res;
}
```

# Project GoodChoice - 필터링 미션

쿼리를 이용한 필터링

\$all -> 배열안의 값이 모두 일치 하는가  
possibleData 안에 roomDuringDay가 전부 있는지

```
// 특정 데이터 검색
exports.searchRooms = (req, res) =>{
  const roomDuringDay = req.body.duringDay;
  const roomMinPrice = req.body.minPrice * 10000;
  const roomMaxPrice = req.body.maxPrice * 10000;
  const roomReservation = req.body.reservation;
  const roomType = req.body.type;
  const roomMaxPersonnel = req.body.people;

  Room.find({
    possibleDate : { $all : roomDuringDay },
    price : { $gte : roomMinPrice, $lte: roomMaxPrice },
    reservation : roomReservation,
    type : { $in : roomType },
    maxPersonnel : { $lte: roomMaxPersonnel },
  }, function(err, docs){
    if(err) return res.status(500).send('Room 검색 실패');
    return res.status(200).send(docs);
  });
}
```

\$gte -> 이상  
\$lte -> 이하

\$in ->  
배열 안의 값 중  
하나라도 일치

# Project GoodChoice - 시연

<http://localhost:3000/>