

# 인사 프로젝트

HR ERP System Presentation

# 목차

01

인사 업무 소개

02

JS와 TS의 차이

03

Axios, Fetch 차이

04

프로젝트 개발환경

05

프로젝트 설계과정

06

Redux-saga, 주요 코드  
설명

07

프로젝트 시연

# 인사(HR)란?

## 사원 등록

신규 입사자의 정보를 시스템에 등록하고,  
인사 서류를 정리하는 과정

## 인사발령

직원의 직책, 부서 이동, 승진 및  
기타 발령 사항을 결정하고 실행

## 인사 고과

직원의 업무 성과와 역량을 평가하여  
보상 및 승진 등의 기준을 마련

## 근무 태도

직원의 근무 성실성, 출퇴근 기록 및 업무 태도를 관리

## 급여 계산 및 지급

근무 시간, 성과, 수당 등을 반영하여 급여를 산정하고 지급



# Javascript, Tyescript

항목	JavaScript (JS)	TypeScript (TS)
언어 유형	동적 타입 (Dynamic Typing)	정적 타입 (Static Typing)
타입 검사	런타임(실행 중) 오류 발견	컴파일 타임에 오류 발견
확장자	.js	.ts
컴파일 과정	필요 없음 (바로 실행 가능)	TS → JS로 변환 필요
브라우저 지원	기본 지원	직접 실행 불가, JS로 변환 후 실행
타입 시스템	타입이 없음 (유형이 동적)	강력한 타입 시스템 제공
코드 안정성	타입 관련 버그 발생 가능	타입 안정성이 높아 오류 감소
객체 지향 지원	ES6 이상에서 지원 (제한적)	클래스, 인터페이스, 제네릭 지원
도구 지원	기본적인 IDE 기능 제공	강력한 코드 자동 완성 및 리팩토링 지원
사용 목적	빠른 개발, 간단한 스크립팅	대규모 프로젝트, 유지보수 용이

# Axios, Fetch

항목	Axios	Fetch
설치 여부	별도로 설치 필요 (npm install axios)	브라우저 내장 API로 별도 설치 필요 없음
기본 제공 기능	자동 JSON 변환, 요청 취소, 인터셉터, 시간 초과 설정 지원	JSON 변환 필요 (.json() 사용), 요청 취소나 시간 초과 직접 구현 필요
응답 처리	응답 데이터 자동 변환 (response.data)	.json() 호출 필요 (response.json())
오류 처리	HTTP 오류 (4xx, 5xx)도 .catch()에서 감지 가능	HTTP 오류는 catch에서 감지되지 않고 response.ok 확인 필요
요청 설정	baseUrl, 인터셉터 등을 통해 요청을 쉽게 관리 가능	fetch(url, { method, headers, body }) 형태로 직접 설정 필요
취소 기능	CancelToken을 제공하여 요청 취소 가능	AbortController를 사용해 직접 구현해야 함
지원 브라우저	대부분의 브라우저 및 Node.js에서 지원	최신 브라우저에서는 지원하지만 일부 구형 브라우저에서는 지원 안 됨

# Axios, Fetch

## Axios 사용 예제

```
import axios from 'axios';

axios.get('https://jsonplaceholder.typicode.com/posts')
  .then(response => console.log(response.data)) // 자동 JSON 변환
  .catch(error => console.error('Error:', error));
```

## Fetch 사용 예제

```
fetch('https://jsonplaceholder.typicode.com/posts')
  .then(response => {
    if (!response.ok) {
      throw new Error('HTTP error! Status: ' + response.status);
    }
    return response.json();
  })
  .then(data => console.log(data))
  .catch(error => console.error('Error:', error));
```



# 개발 환경

Front-end

Back-end



Spring  
Boot 3



Java™



DB/Server



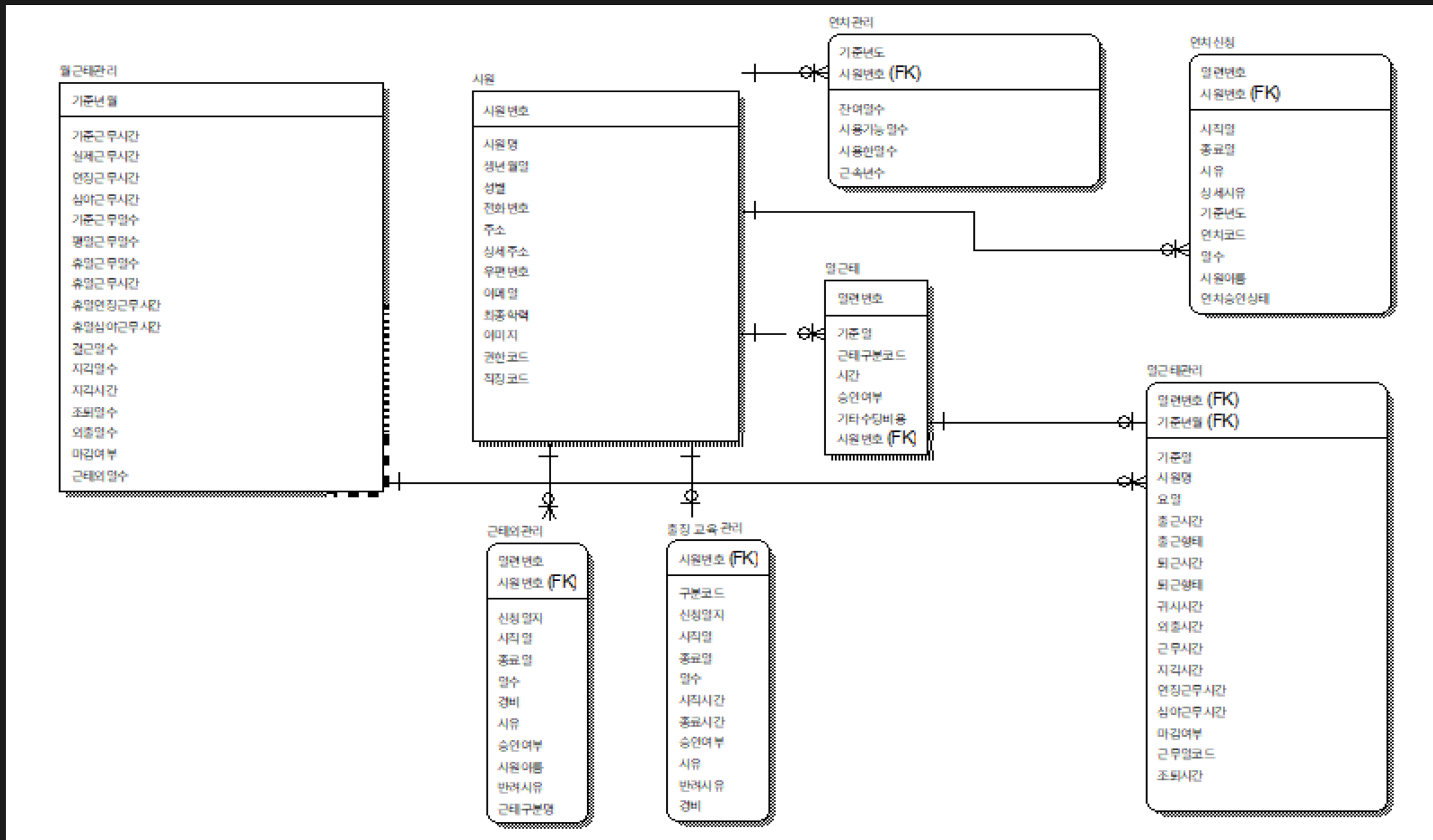
Tools/Etc



# 프로젝트 선계개저

## 데이터 모델링

업무 요구사항을 분석하여 데이터베이스 구조를 설계하는 과정





# 코드 설명

## Redux-saga

단계	Redux 기본 흐름	Redux-Saga 흐름
1	컴포넌트에서 액션을 디스패치(dispatch)	컴포넌트에서 액션을 디스패치(dispatch)
2	리듀서(reducer)에서 상태(state)를 업데이트	Saga에서 비동기 작업을 수행한 후 액션을 디스패치
3	변경된 상태를 UI에 반영	리듀서에서 상태(state)를 업데이트
4	-	변경된 상태를 UI에 반영

# 코드 설명

## Redux-saga

```
// 사원 조회
function* fetchEmpList() {
  try {
    const response: AxiosResponse = yield call(getEmplist);
    console.log('사원 조회 response', response);
    yield put(attdActions.getEmpListSuccess(response.data.list));
  } catch (error: any) {
    console.log(error);
  }
}
```

# 코드 설명

## Redux-saga

```
// 연차 신청
function* registBreakAttd(action: any) {
  const { payload } = action;
  try {
    const response: AxiosResponse = yield call(insertBreaKAttd, payload);
    console.log('신청 response', response);
  } catch (error: any) {
    console.log(error);
  }
}
```

# 코드 설명

## Redux-saga

```
export function* watchAttdActions() {
  yield takeLatest(attdActions.getEmpListRequest, fetchEmpList);
  yield takeLatest(attdActions.getRestAttdListRequest, fetchRestAttdList);
  yield takeLatest(attdActions.registRestAttdRequest, registRestAttd);
  yield takeLatest(attdActions.registRestAttdRequest2, registRestAttd2);
  yield takeLatest(attdActions.approvalRestAttdRequest, approvalRestAttdList);
  yield takeLatest(attdActions.removeRestAttdRequest, removeRestAttdList);
  yield takeLatest(attdActions.getBreakAttdListRequest, fetchBreakAttdList);
  yield takeLatest(attdActions.registBreakAttdRequest, registBreakAttd);
  yield takeLatest(attdActions.approvalBreakAttdRequest, approvalBreakAttdList);
  yield takeLatest(attdActions.removeBreakAttdRequest, removeBreakAttdList);
}

export default function* attdSaga() {
  yield all([fork(watchAttdActions)]);
}
```



# 코드 설명

## 주요 코드 분석

🕒 근태 관리 ^

📍 근태 신청 및 등록 ^


- 일근태 등록
- 근태외 신청
- 연차 신청
- 출장/교육 신청
- 초과근무 신청

👁 근태 승인 v

📁 급여 관리 v

📅 교육 관리 v

연차신청



ⓘ Image size Limit should be 125kb Max.

연차구분

오전반차

신청자

진양철

시작날짜

2025-02-14

종료날짜

2025-02-14

시작시간

오전 10:41

종료시간

오전 10:47

일수

0.5

사유

타그

일수계산

등록

# 코드 설명

## 주요 코드 분석

```
//연차 신청을 하기위해 restAttdTO에 데이터를 담아 api로 보냄
dispatch(attdActions.registBreakAttdRequest(restAttdTO));

alert('신청이 완료 되었습니다.');
```

```
window.location.reload();
};

/* 일수 계산 함수 */
function calculateNumberOfDays() {
  const startMs = Number(new Date(startDate).getTime());
  //startMs 구하면 밀리초(milliseconds)로 나오게됨.
  console.log('startMs는 몇시', startMs);
  const endMs = Number(new Date(endDate).getTime());
  if (attdType === '오전반차' || attdType === '오후반차') setNumberOfDays(0.5);
  //시간이 밀리초로 나오기 때문에 1000밀리초 * 60초 * 60분 * 24시로 구하게됨
  // +1을 하는 이유는 첫날과 마지막 날을 포함하여 계산하기 때문
  else setNumberOfDays((endMs - startMs) / (1000 * 60 * 60 * 24) + 1);
}
```



# 코드 설명

## 주요 코드 분석

승인완료

반려

취소

삭제

```
// 연차 승인/반려
const handleUpdateExcusedAttd = (sendData: any) => {
  try {
    dispatch(attdActions.approvalBreakAttdRequest(sendData));

    console.log('데이터 업데이트 완료');
    setSelectRow([]);
  } catch (error) {
    console.error('데이터 업데이트 오류:', error);
  }
};
```

# 코드 설명

## 주요 코드 분석

### 교육 신청

신청자 진양철	교육과목 부
시작날짜 연도-월-일	종료날짜 연도-월-일
시작시간 --:--:--	종료시간 --:--:--
사유	
<input type="text"/>	
<input type="button" value="신청"/>	

// 교육 리스트 세팅

```
useEffect(() => {  
  dispatch(attdActions.getEmpListRequest());  
  
  const today = new Date();  
  const year = today.getFullYear();  
  const month = String(today.getMonth() + 1).padStart(2, '0');  
  const day = String(today.getDate()).padStart(2, '0');  
  setRequestDate(`${year}-${month}-${day}`);  
}, []);
```

```
useEffect(() => {  
  setEmpList(rawList);  
}, [rawList]);
```

```
const empLists = (empList ?? []).map((item) => {  
  if (item.empCode === localStorage.getItem('empCode'))  
    return (  
      <MenuItem value={item.empCode} key={item.empCode}>  
        {item.empName}  
      </MenuItem>  
    );  
});
```

# 코드 설명

## 주요 코드 분석

### 교육 신청

신청자 진양철	교육과목 부
시작날짜 연도-월-일	종료날짜 연도-월-일
시작시간 --:--:--	종료시간 --:--:--
사유	
<input type="text"/>	
<input type="button" value="신청"/>	

// 교육 리스트 세팅

```
useEffect(() => {  
  dispatch(attdActions.getEmpListRequest());  
  
  const today = new Date();  
  const year = today.getFullYear();  
  const month = String(today.getMonth() + 1).padStart(2, '0');  
  const day = String(today.getDate()).padStart(2, '0');  
  setRequestDate(`${year}-${month}-${day}`);  
}, []);
```

```
useEffect(() => {  
  setEmpList(rawList);  
}, [rawList]);
```

```
const empLists = (empList ?? []).map((item) => {  
  if (item.empCode === localStorage.getItem('empCode'))  
    return (  
      <MenuItem value={item.empCode} key={item.empCode}>  
        {item.empName}  
      </MenuItem>  
    );  
});
```