



Contents•

01 팀원 소개 3p

Introduction to team members

04 설계 12p

server design

02 프로젝트 소개 6p

Introduction to projects

05 화면 UI 및 시연 64p

Screen UI and demonstration

03 개발 일정 9p

Development Schedule

06 후기 | 문제해결 100p

Reviews and troubleshooting

01

팀원 소개 ·

팀원소개



팀장 정선희

- Data Base ✓
- SQL ✓
- UI/UX ✘
- JSP ✓
- JS/JQ ✘
- CSS ✓
- Test ✓
- Debugging ✓



부팀장 안임마누엘

- Data Base ✓
- SQL ✓
- UI/UX ✘
- JSP ✓
- JS/JQ ✘
- CSS ✓
- Test ✓
- Debugging ✓



팀원 안소민

- Data Base ✓
- SQL ✓
- UI/UX ✘
- JSP ✓
- JS/JQ ✘
- CSS ✓
- Test ✓
- Debugging ✓



팀원 김명균

- Data Base ✓
- SQL ✓
- UI/UX ✘
- JSP ✓
- JS/JQ ✘
- CSS ✓
- Test ✓
- Debugging ✓

팀원소개



팀원 김정학

- Data Base
- SQL
- UI/UX
- JSP
- JS/JQ
- CSS
- Test
- Debugging



팀원 신근수

- Data Base
- SQL
- UI/UX
- JSP
- JS/JQ
- CSS
- Test
- Debugging



팀원 채병훈

- Data Base
- SQL
- UI/UX
- JSP
- JS/JQ
- CSS
- Test
- Debugging

02

프로젝트 소개[.]

프로젝트 소개

ABOUT OUR PROJECT

WEB 기반 프로젝트 파견 관리 & 보고서 결재 **HR 솔루션**

개발자 및 프리랜서의 상세 정보 및
프로젝트 파견현황
실시간 조회

프로젝트 현황을 실시간으로 확인하고
프로젝트에 필요한 인원을 직접 찾을 수 있는
편리함 제공

기존 서류 결재를 온라인화 함으로써
업무 진행의 검토와 승인 절차의
효율성 극대화

적용기술 | 개발환경

보유기술

Java

Spring

SQL

Jquery

ORACLE 11g

JavaScript

JSP[EL, JSTL, Directive 외]

CSS 3.0

HTML 5

개발도구

(STS) SpringToll Suite 4.17.0

SQL Gate

DBeaver

Notepad++

IntelliJ

VScode

개발환경

Spring boot

Spring framework 2.7.7

Apache Tomcat

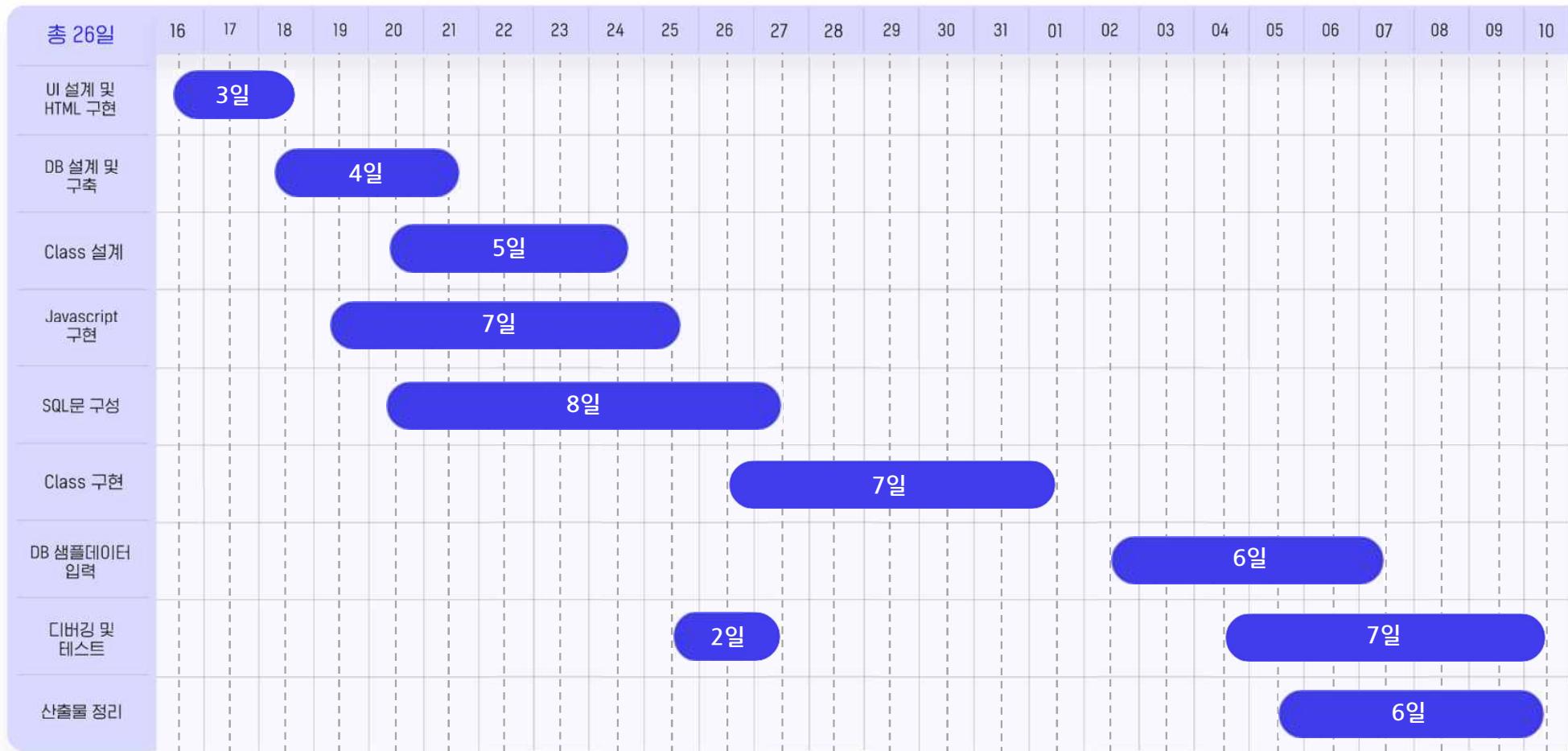
JDK 1.8

Windows 10 Professional

03

개발일정.

개발일정



회의록

1주차

UI설계 및 HTML 구현, DB 설계 및 구축 / 파트 배분

DB 설계 인원 3명 UI 설계 및 HTML 구현 인원 4명 각 페이지 담당

! 주요 안건

- 기한 준수
- name 값 약속
- 파트별 역할 명확히 정하기

2주차

Class 설계 / 입력 SQL문 구성 / HTML을 바탕으로 Javascript 구축 / 파트 배분

SQL[mapper] 담당: 3명/JAVA Class 담당: 2명/JS 담당: 2명

! 주요 안건

- DTO의 name 값 일관성 유지
- 쿼리문 구상
- mapper id 값 JAVA와 일치시키기

3주차

JavaScript 구현 / Class 구현 / 수정, 삭제 SQL 문 구성 / DB 연동 / Class, JS 합침 / 오류 대응

소스 병합: 1명 / SQL[mapper]: 2명 / JavaScript: 2명 / JAVA Class: 2명

! 주요 안건

- 기한 미준수에 대한 조치
- 자바스크립트 사용할 함수, 메소드 미리 정리
- 수정, 삭제 SQL 구문 추가

4주차

오류 수정 / 유효성 검사 및 오류 수정 / 검색 SQL 문 구성 / DB 셱플 데이터 입력 / 테스트 / CSS 추가

소스 병합 및 오류수정: 2명 / 샘플 데이터 입력: 2명 / 테스트: 2명 / CSS: 1명

! 주요 안건

- 파일 합치면서 발생하는 오류 기록 및 수정
- CSS 구상 및 시작
- 샘플데이터 입력 시작
- 검색 SQL 구문 추가
- 메인화면 시각적인 컨텐츠 추가

5주차

총합 테스트 / 오류 수정 / 추가 DB 셱플데이터 입력 / 테스트 / CSS 추가

오류수정: 2명 / 테스트: 2명 / 샘플 데이터: 2명 / CSS: 1명

! 주요 안건

- 테스트 진행
- 추가 샘플 데이터 입력
- 오류 기록 및 수정
- CSS 추가

6주차

총합 테스트 / PPT 작업 및 정리 / 프로젝트 후기 작성

! 주요 안건

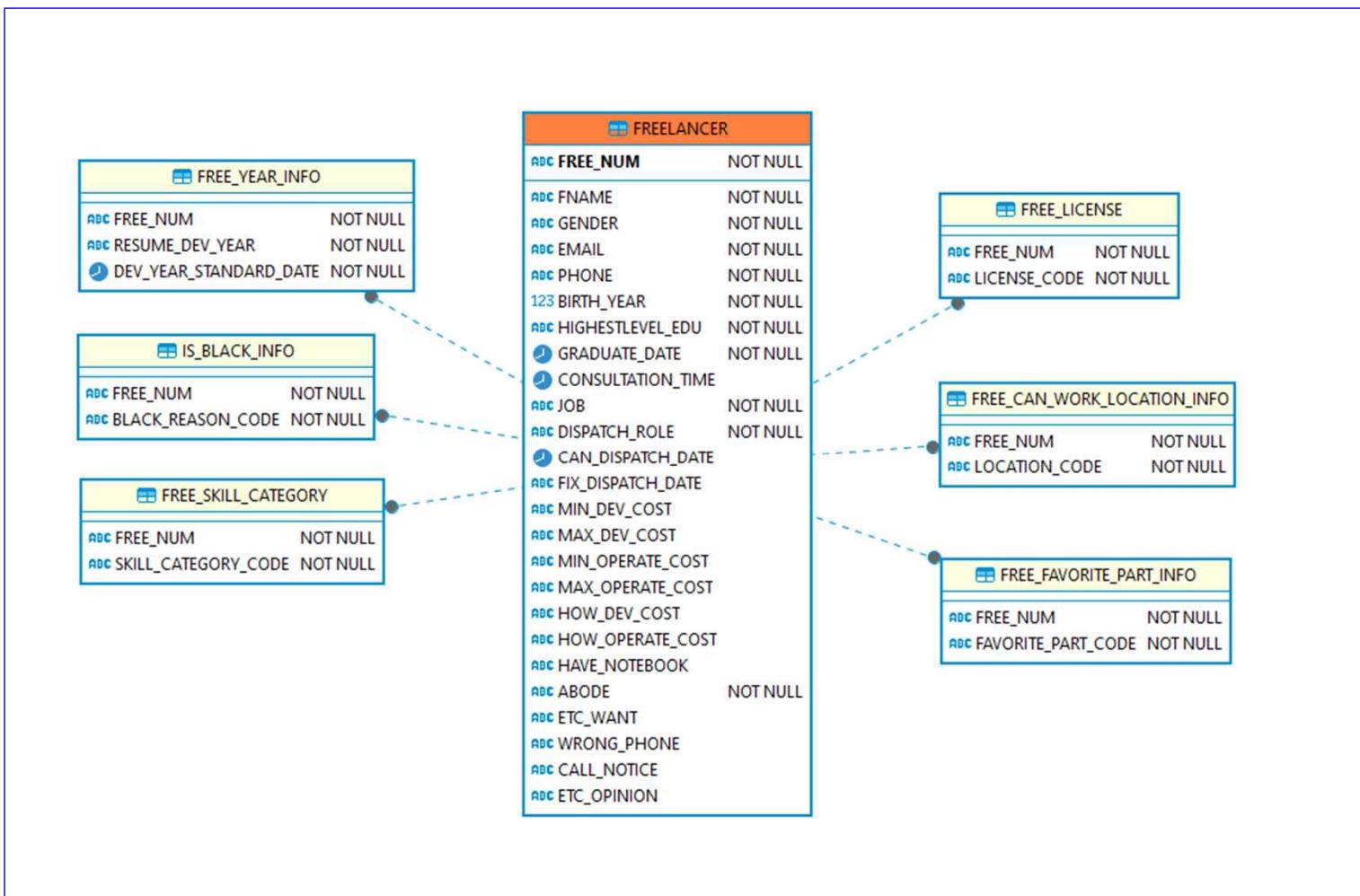
- PPT 작업 시작
- 프로젝트 후기 작성

04

테이블 설계

DB테이블 설계 • 프리랜서

테이블 총 7개



DB테이블 설계 • 프리랜서

프리랜서 정보

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREELANCER	FREE_NUM	프리랜서정보	프리랜서 번호	Y			VARCHAR2(10)		
	FNAME		프리랜서 이름				VARCHAR2(30)		
	GENDER		성별				VARCHAR2(2)		
	EMAIL		이메일				VARCHAR2(30)		
	PHONE		전화번호				VARCHAR2(11)		
	BIRTH_YEAR		출생년도				NUMBER(38,1)		
	HIGHESTLEVEL_EDU		최종학력				VARCHAR2(15)		CODE_HIGHESTLEVEL_EDU
	GRADUATE_DATE		졸업년도				DATE		
	CONSULTATION_TIME		상담시각				DATE		
	JOB		현재 직업				VARCHAR2(10)		CODE_JOB
	DISPATCH_ROLE		파견 역할				VARCHAR2(10)		CODE_DISPATCH_ROLE
	CAN_DISPATCH_DATE		파견 가능일				DATE		
	FIX_DISPATCH_DATE		파견 가능일 확정				VARCHAR2(4)		
	MIN_DEV_COST		최소 개발 단가				VARCHAR2(10)		
	MAX_DEV_COST		최대 개발 단가				VARCHAR2(10)		
	MIN_OPERATE_COST		최소 운영 단가				VARCHAR2(10)		
	MAX_OPERATE_COST		최대 운영 단가				VARCHAR2(10)		
	HOW_DEV_COST		개발 단가 고정/협상 후 결정				VARCHAR2(10)		
	HOW_OPERATE_COST		운영 단가 고정/협상 후 결정				VARCHAR2(10)		
	HAVE_NOTEBOOK		노트북 소유				VARCHAR2(10)		
	ABODE		거주지				VARCHAR2(10)		
	ETC_WANT		기타 요구사항				VARCHAR2(600)		
	WRONG_PHONE		전화번호 틀림				VARCHAR2(10)		
	CALL_NOTICE		전화상담 시 주의사항				VARCHAR2(600)		
	ETC_OPINION		기타 평가				VARCHAR2(600)		

DB테이블 설계 • 프리랜서

프리랜서 근무 가능 지역

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREE_CAN_WORK_LOCATION_INFO	FREE_NUM LOCATION_CODE	근무 가능 지역	프리랜서 번호 근무 가능 지역		Y Y		VARCHAR2(10) VARCHAR2(10)		FREELANCER CODE_LOCATION

프리랜서 선호 분야

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREE_FAVORITE_PART_INFO	FREE_NUM FAVORITE_PART_CODE	선호 분야	프리랜서 번호 선호 분야		Y Y		VARCHAR2(10) VARCHAR2(10)		FREELANCER CODE_FAVORITE_PART

프리랜서 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREE_LICENSE	FREE_NUM LICENSE_CODE	자격증	프리랜서 번호 자격증		Y Y		VARCHAR2(10) VARCHAR2(10)		FREELANCER CODE_LICENSE

DB테이블 설계 • 프리랜서

프리랜서 가상 개발년차

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREE_YEAR_INFO	FREE_NUM	가상 개발년차	프리랜서 번호		Y		VARCHAR2(10)		FREELANCER
	RESUME_DEV_YEAR		추정 개발년차				VARCHAR2(30)		
	DEV_YEAR_STANDARD_DATE		이력서 상 개발년차				DATE		

프리랜서 블랙 개발자

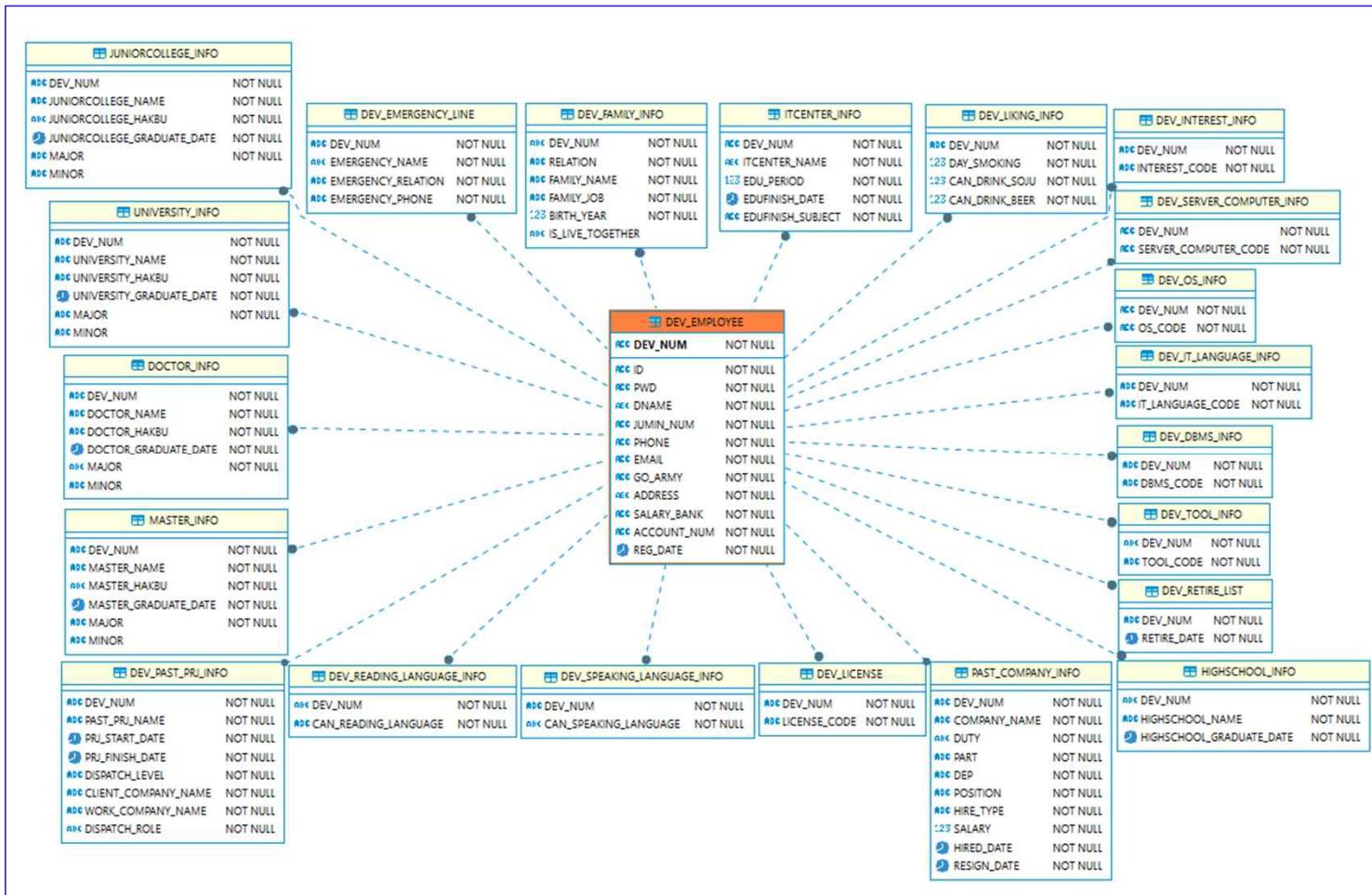
물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
IS_BLACKINFO	FREE_NUM	블랙개발자	프리랜서 번호		Y		VARCHAR2(10)		FREELANCER
	BLACK_REASON_CODE		블랙개발자 번호		Y		VARCHAR2(10)		CODE_BLACK_REASON

프리랜서 스킬 분야

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
FREE_SKILL_CATEGORY	FREE_NUM	스킬분야	프리랜서 번호		Y		VARCHAR2(10)		FREELANCER
	SKILL_CATEGORY_CODE		스킬분야		Y		VARCHAR2(10)		CODE_SKILL_CATEGORY

DB테이블 설계 정규직

테이블 총 22개



DB테이블 설계 정규직

정규직 정보

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_EMPLOYEE	DEV_NUM	정규직정보	정규직 번호	Y			VARCHAR2(10)		
	ID		아이디				VARCHAR2(20)		
	PWD		비밀번호				VARCHAR2(20)		
	DNAME		정규직 이름				VARCHAR2(30)		
	JUMIN_NUM		주민번호				VARCHAR2(14)		
	PHONE		전화번호				VARCHAR2(11)		
	EMAIL		이메일				VARCHAR2(30)		
	GO_ARMY		병역		Y		VARCHAR2(10)		CODE_GO_ARMY
	ADDRESS		주소				VARCHAR2(100)		
	SALARY_BANK		은행이름		Y		VARCHAR2(30)		CODE_SALARY_BANK
	ACCOUNT_NUM		계좌번호				VARCHAR2(30)		
	REG_DATE		입사일				DATE	SYSDATE	

정규직 비상 연락망

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_EMERGENCY_LINE	DEV_NUM	정규직 비상연락망	정규직 번호	Y			VARCHAR2(10)		DEV_EMPLOYEE
	EMERGENCY_NAME		비상시 연락할 사람 이름				VARCHAR2(30)		
	EMERGENCY_RELATION		연락할 사람과의 관계				VARCHAR2(10)		
	EMERGENCY_PHONE		연락할 사람의 번호				VARCHAR2(11)		

정규직 가족관계

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_FAMILY_INFO	DEV_NUM	정규직 가족 관계	정규직 번호	Y			VARCHAR2(10)		DEV_EMPLOYEE
	RELATION		관계				VARCHAR2(12)		
	FAMILY_NAME		가족 이름				VARCHAR2(40)		
	FAMILY_JOB		가족의 직업				VARCHAR2(40)		
	BIRTH_YEAR		가족의 생년월일				NUMBER(9)		
	IS_LIVE_TOGETHER		동거 여부			Y	VARCHAR2(10)	'비동거'	

DB테이블 설계 정규직

정규직 개발 경력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_PAST_PRJ_INFO	DEV_NUM	정규직 회사 경력	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	PAST_PRJ_NAME		프로젝트명				VARCHAR2(60)		
	PRJ_START_DATE		시작일				DATE		
	PRJ_FINISH_DATE		종료일				DATE		
	DISPATCH_LEVEL		파견 단계				VARCHAR2(10)		
	CLIENT_COMPANY_NAME		고객사				VARCHAR2(60)		
	WORK_COMPANY_NAME		근무사				VARCHAR2(60)		
	DISPATCH_ROLE		역할				VARCHAR2(50)		

정규직 개발 경력 사용 기종

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_SERVER_COMPUTER_INFO	DEV_NUM	사용 기종	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	SERVER_COMPUTER_CODE		기종		Y		VARCHAR2(10)		CODE_SERVER_COMPUTER

정규직 개발 경력 사용 OS

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_OS_INFO	DEV_NUM	사용 OS	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	OS_CODE		OS		Y		VARCHAR2(10)		CODE_OS

정규직 개발 경력 사용 언어

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_IT_LANGUAGE_INFO	DEV_NUM	사용 언어	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	IT_LANGUAGE_CODE		사용 언어		Y		VARCHAR2(10)		CODE_IT_LANGUAGE

DB테이블 설계 • 정규직

정규직 개발 경력 사용 DBMS

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_DBMS_INFO	DEV_NUM DBMS_CODE	사용 DBMS	정규직 번호 DBMS		Y Y		VARCHAR2(10) VARCHAR2(10)		DEV_EMPLOYEE CODE_DBMS

정규직 개발 경력 사용 TOOL

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_TOOL_INFO	DEV_NUM TOOL_CODE	사용 TOOL	정규직 번호 TOOL		Y Y		VARCHAR2(10) VARCHAR2(10)		DEV_EMPLOYEE CODE_TOOL

정규직 관심분야

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_INTEREST_INFO	DEV_NUM INTEREST_CODE	관심분야	정규직 번호 관심분야		Y Y		VARCHAR2(10) VARCHAR2(10)		DEV_EMPLOYEE CODE_INTEREST

정규직 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_LICENSE	DEV_NUM LICENSE_CODE	자격증	정규직 번호 자격증		Y Y		VARCHAR2(10) VARCHAR2(10)		DEV_EMPLOYEE CODE_LICENSE

DB테이블 설계 • 정규직

정규직 기호

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_LIKING_INFO	DEV_NUM	기호	정규직 번호			Y	VARCHAR2(10)		DEV_EMPLOYEE
	DAY_SMOKING		하루 흡연량				NUMBER(2)		
	CAN_DRINKING_SOJU		소주 주량				NUMBER(2)		
	CAN_DRINKING_BEER		맥주 주량				NUMBER(2)		

정규직 독해 가능 외국어

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_READING_LANGUAGE_INFO	DEV_NUM	독해 가능 외국어	정규직 번호			Y	VARCHAR2(10)		DEV_EMPLOYEE
	CAN_READING_LANGUGAE		독해 가능 외국어			Y	VARCHAR2(10)		CODE_FOREIGN_LANGUAGE

정규직 회화 가능 외국어

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DEV_SPEAKING_LANGUAGE_INFO	DEV_NUM	회화 가능 외국어	정규직 번호			Y	VARCHAR2(10)		DEV_EMPLOYEE
	CAN_SPEAKING_LANGUGAE		회화 가능 외국어			Y	VARCHAR2(10)		CODE_FOREIGN_LANGUAGE

DB테이블 설계 • 정규직

정규직 고등학교

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
HIGHSCOOL_INFO	DEV_NUM	고등학교 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	HIGHSCHOOL_NAME		고등학교 정보				VARCHAR2(40)		
	HIGHSCHOOL_GRADUATE_DATE		졸업년도				DATE		

정규직 전문대학교

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
JUNIORCOLLEGE_INFO	DEV_NUM	전문대 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	JUNIORCOLLEGE_NAME		전문대 이름				VARCHAR2(40)		
	JUNIORCOLLEGE_HAKBU		전문대 학부		Y		VARCHAR2(12)		CODE_HAKBU
	JUNIORCOLLEGE_GRADUATE_DATE		전문대 졸업년도				DATE		
	MAJOR		전문대 전공 과목				VARCHAR2(40)		
	MINOR		전문대 부전공 과목			Y	VARCHAR2(40)		

정규직 대학교

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
UNIVERSITY_INFO	DEV_NUM	대학교 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	UNIVERSITY_NAME		대학교 이름				VARCHAR2(40)		
	UNIVERSITY_HAKBU		대학교 학부		Y		VARCHAR2(12)		CODE_HAKBU
	UNIVERSITY_GRADUATE_DATE		대학교 졸업년도				DATE		
	MAJOR		대학교 전공 과목				VARCHAR2(40)		
	MINOR		대학교 부전공 과목			Y	VARCHAR2(40)		

DB테이블 설계 정규직

정규직 대학교 석사

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
MASTER_INFO	DEV_NUM	대학교 석사 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	MASTER_NAME		대학교 석사 이름				VARCHAR2(40)		
	MASTER_HAKBU		대학교 석사 학부		Y		VARCHAR2(12)		CODE_HAKBU
	MASTER_GRADUATE_DATE		대학교 석사 졸업년도				DATE		
	MAJOR		대학교 석사 전공 과목				VARCHAR2(40)		
	MINOR		대학교 석사 부전공 과목		Y		VARCHAR2(40)		

정규직 대학교 박사

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
DOCTOR_INFO	DEV_NUM	대학교 박사 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	DOCTER_NAME		대학교 박사 이름				VARCHAR2(40)		
	DOCTER_HAKBU		대학교 박사 학부		Y		VARCHAR2(12)		CODE_HAKBU
	DOCTOR_GRADUATE_DATE		대학교 박사 졸업년도				DATE		
	MAJOR		대학교 박사 전공 과목				VARCHAR2(40)		
	MINOR		대학교 박사 부전공 과목		Y		VARCHAR2(40)		

정규직 IT 교육센터

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
ITCENTER_INFO	DEV_NUM	IT 교육센터	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	ITCENTER_NAME		IT교육센터 이름				VARCHAR2(60)		
	EDU_PERIOD		교육 기간				NUMBER(10)		
	EDUFINISH_DATE		수료 날짜				DATE		
	EDUFINISH SUBJECT		수료 과목				VARCHAR2(40)		

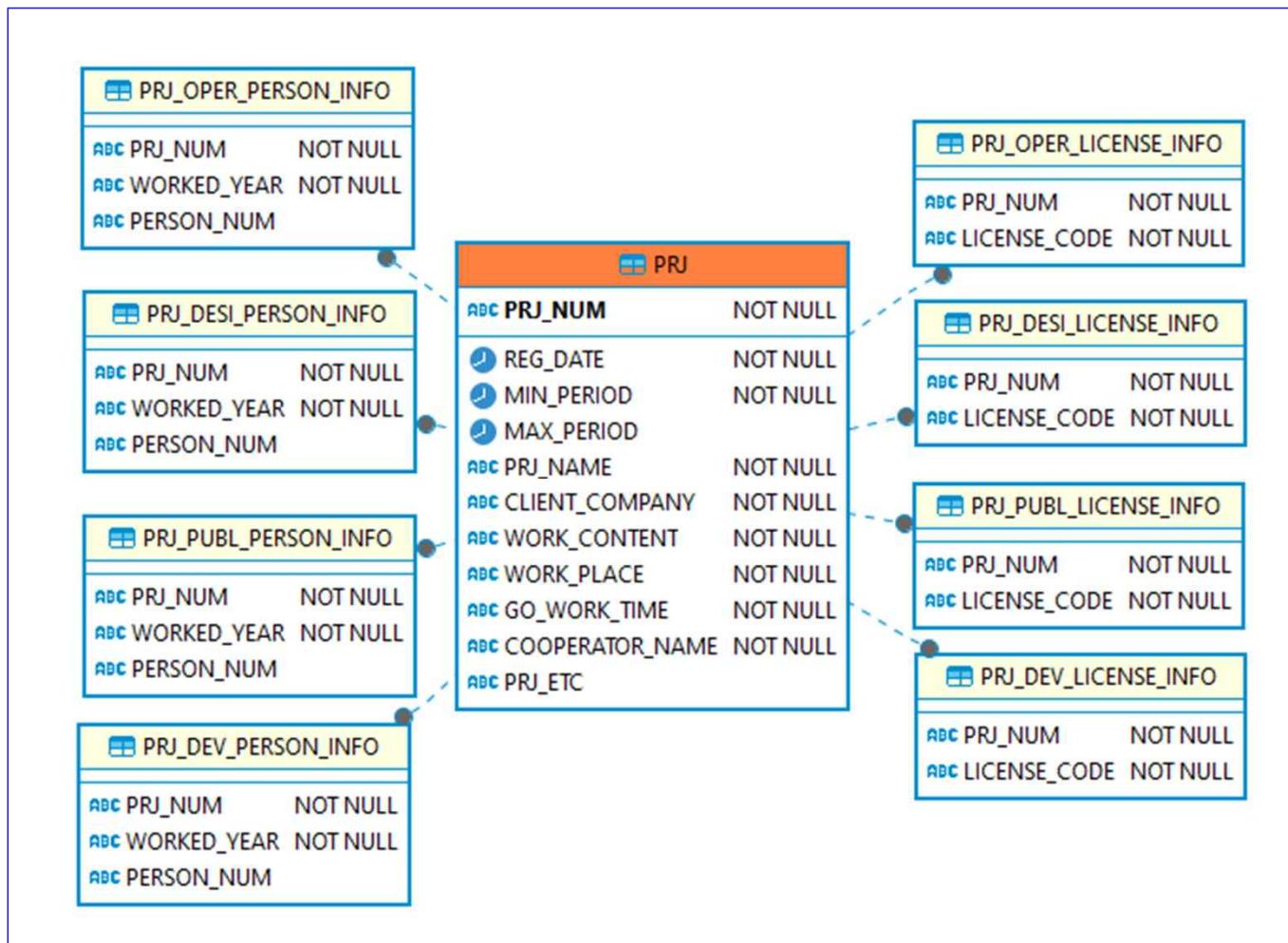
DB테이블 설계 • 정규직

정규직 회사 경력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PAST_COMPANY_INFO	DEV_NUM	회사 경력 정보	정규직 번호		Y		VARCHAR2(10)		DEV_EMPLOYEE
	COMPANY_NAME		회사명				VARCHAR2(60)		
	DUTY		업무		Y		VARCHAR2(10)		CODE_DUTY
	PART		분야		Y		VARCHAR2(10)		CODE_PART
	DEP		부서		Y		VARCHAR2(10)		CODE_DEP
	POSITION		직위		Y		VARCHAR2(10)		CODE_POSITION
	HIRE_TYPE		고용형태		Y		VARCHAR2(10)		CODE_HIRE_TYPE
	SALARY		연봉				NUMBER(5)		
	HIRE_DATE		입사일				DATE		
	RESIGN_DATE		퇴사일				DATE		

DB테이블 설계•프로젝트

테이블 총 9개



DB테이블 설계 • 프로젝트

프로젝트 정보

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ	PRJ_NUM	프로젝트	프로젝트 번호	Y			VARCHAR2(10)		
	REG_DATE		접수일자				DATE		
	MIN_PERIOD		최소 기간				DATE		
	MAX_PERIOD		최대 기간			Y	DATE		
	PRJ_NAME		프로젝트 명				VARCHAR2(60)		
	CLIENT_COMPANY		고객사				VARCHAR2(60)		
	WORK_CONTENT		업무내용				VARCHAR2(60)		
	WORK_PLACE		업무장소				VARCHAR2(60)		
	GO_WORK_TIME		출근시간				VARCHAR2(4)		
	COOPERATOR_NAME		협력업체				VARCHAR2(60)		
	PRJ_ETC		기타			Y	VARCHAR2(600)		

프로젝트 개발 인력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_DEV_PERSON_INFO	PRJ_NUM	프로젝트 개발 인력	프로젝트 번호	Y			VARCHAR2(10)		PRJ
	WORKED_YEAR		년차				VARCHAR2(10)		
	PERSON_NUM		개발 사람 수		Y		VARCHAR2(10)		

프로젝트 개발 인력 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_DEV_LICENSE	PRJ_NUM	개발 인력 자격증	프로젝트 번호	Y			VARCHAR2(10)		PRJ
	LICENSE_CODE		자격증		Y		VARCHAR2(10)		CODE_LICENSE

DB테이블 설계•프로젝트

프로젝트 운영 인력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_OPER_PERSON_INFO	PRJ_NUM	프로젝트 운영 인력	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	WORKED_YEAR		년차				VARCHAR2(10)		
	PERSON_NUM		운영 사람 수			Y	VARCHAR2(10)		

프로젝트 운영 인력 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_OPER_LICENSE	PRJ_NUM	운영 인력 자격증	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	LICENSE_CODE		자격증		Y		VARCHAR2(10)		CODE_LICENSE

프로젝트 퍼블리셔 인력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_PUBL_PERSON_INFO	PRJ_NUM	프로젝트 퍼블리셔 인력	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	WORKED_YEAR		년차				VARCHAR2(10)		
	PERSON_NUM		운영 사람 수			Y	VARCHAR2(10)		

DB테이블 설계•프로젝트

프로젝트 디자인 입력

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_DESI_PERSON_INFO	PRJ_NUM	디자인 운영 인력	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	WORKED_YEAR		년차				VARCHAR2(10)		
	PERSON_NUM		운영 사람 수			Y	VARCHAR2(10)		

프로젝트 디자인 입력 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_DESI_LICENSE	PRJ_NUM	디자인 인력 자격증	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	LICENSE_CODE		자격증		Y		VARCHAR2(10)		CODE_LICENSE

프로젝트 퍼블리셔 입력 자격증

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_PUBL_LICENSE	PRJ_NUM	퍼블리셔 인력 자격증	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	LICENSE_CODE		자격증		Y		VARCHAR2(10)		CODE_LICENSE

DB테이블 설계•결제 | 공지사항 | 파견

테이블 총 1개

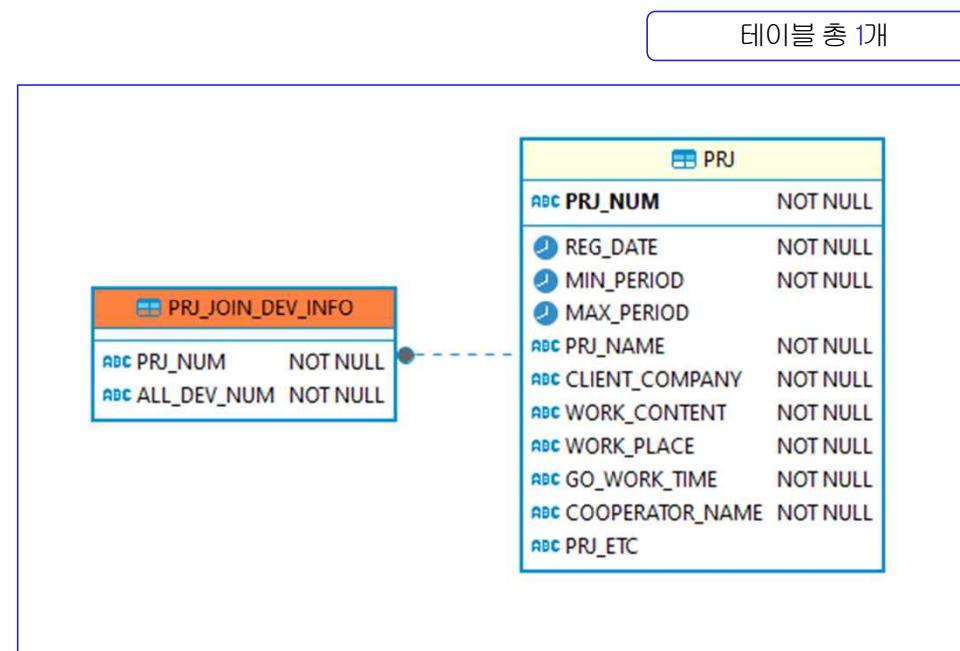
GYEOLJAE	
GYEOLJAE_NUM	NOT NULL
REG_DATE	NOT NULL
WRITER	NOT NULL
TITLE	NOT NULL
CONTENT	NOT NULL
DOCU_TYPE	NOT NULL
GYEOLJAE_PROGRESS	
GYEOLJAE_PROGRESS_DATE	

DB테이블 설계 • 결제 | 공지사항 | 파견

테이블 총 1개

BOARD	
123	B_NO
abc	SUBJECT
abc	WRITER
abc	CONTENT
⌚	REG_DATE
123	READCOUNT
abc	IMG_NAME

DB테이블 설계•결제 | 공지사항 | 파견



DB테이블 설계 • 결제 | 공지사항 | 파견

결재 정보

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
GYEOLJAE	GYEOLJAE_NUM	결재 정보	결재 번호	Y			VARCHAR2(10)		
	REG_DATE		제출 날짜				DATE	SYSDATE	
	WRITER		작성자		Y		VARCHAR2(30)		DEV_EMPLOYEE
	TITLE		제목				VARCHAR2(200)		
	CONTENT		내용				VARCHAR2(4000)		
	DOCU_TYPE		문서 종류				VARCHAR2(30)		
	GYEOLJAE_PROGRESS		결재 처리 상태			Y	VARCHAR2(10)	'미결'	
	GYEOLJAE_PROGRESS_DATE		결재 처리 날짜			Y	DATE		

공지사항 정보

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
BOARD	B_NO	공지사항 정보	공지사항 번호	Y			NUMBER(20)		
	SUBJECT		제목				VARCHAR2(200)		
	WRITER		작성자				VARCHAR2(100)		
	CONTENT		내용				VARCHAR2(3000)		
	REG_DATE		작성 일자				DATE	SYSDATE	
	READCOUNT		조회수				NUMBER(5)		
	IMG_NAME		파일 이름			Y	VARCHAR2(100)		

프로젝트 파견

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
PRJ_JOIN_DEV_INFO	PRJ_NUM	파견	프로젝트 번호		Y		VARCHAR2(10)		PRJ
	ALL_DEV_NUM		파견 번호				VARCHAR2(10)		

DB테이블 설계 참조

테이블 총 26개

KING.CODE_BLACK_REASON	KING.CODE_COOPERATOR_NAME	KING.CODE_DBMS	KING.CODE_DEP	KING.CODE_DISPATCH_LEVEL
<code>RKC BLACK_REASON_CODI NOT NULL</code>	<code>RKC COOPERATOR_NAME_CODI NOT NULL</code>	<code>RKC DBMS_CODI NOT NULL</code>	<code>RKC DEP_CODI NOT NULL</code>	<code>RKC DISPATCH_LEVEL_CODI NOT NULL</code>
<code>RKC BLACK_REASON_NAME NOT NULL</code>	<code>RKC COOPERATOR_NAME NOT NULL</code>	<code>RKC DBMS_NAME NOT NULL</code>	<code>RKC DEP_NAME NOT NULL</code>	<code>RKC DISPATCH_LEVEL_NAME NOT NULL</code>
KING.CODE_DISPATCH_ROLE	KING.CODE_DUTY	KING.CODE_FAVORITE_PART	KING.CODE_FOREIGN_LANGUAGE	KING.CODE_GO_ARMY
<code>RKC DISPATCH_ROLE_CODI NOT NULL</code>	<code>RKC DUTY_CODI NOT NULL</code>	<code>RKC FAVORITE_PART_CODI NOT NULL</code>	<code>RKC FOREIGN_LANGUAGE_CODI NOT NULL</code>	<code>RKC GO_ARMY_CODE NOT NULL</code>
<code>RKC DISPATCH_ROLE_NAME NOT NULL</code>	<code>RKC DUTY_NAME NOT NULL</code>	<code>RKC FAVORITE_PART_NAME NOT NULL</code>	<code>RKC FOREIGN_LANGUAGE_NAME NOT NULL</code>	<code>RKC GO_ARMY_NAME NOT NULL</code>
KING.CODE_HAKBU	KING.CODE_HIGHESTLEVEL_EDU	KING.CODE_HIRE_TYPE	KING.CODE_INTEREST	KING.CODE_IT_LANGUAGE
<code>RKC HAKBU_CODI NOT NULL</code>	<code>RKC HIGHESTLEVEL_EDU_CODI NOT NULL</code>	<code>RKC HIRE_TYPE_CODI NOT NULL</code>	<code>RKC INTEREST_CODI NOT NULL</code>	<code>RKC IT_LANGUAGE_CODI NOT NULL</code>
<code>RKC HAKBU_NAME NOT NULL</code>	<code>RKC HIGHESTLEVEL_EDU_NAME NOT NULL</code>	<code>RKC HIRE_TYPE_NAME NOT NULL</code>	<code>RKC INTEREST_NAME NOT NULL</code>	<code>RKC IT_LANGUAGE_NAME NOT NULL</code>
KING.CODE_JOB	KING.CODE_LICENSE	KING.CODE_LOCATION	KING.CODE_OS	KING.CODE_PART
<code>RKC JOB_CODI NOT NULL</code>	<code>RKC LICENSE_CODI NOT NULL</code>	<code>RKC LOCATION_CODI NOT NULL</code>	<code>RKC OS_CODI NOT NULL</code>	<code>RKC PART_CODI NOT NULL</code>
<code>RKC JOB_NAME NOT NULL</code>	<code>RKC LICENSE_NAME NOT NULL</code>	<code>RKC LOCATION_NAME NOT NULL</code>	<code>RKC OS_NAME NOT NULL</code>	<code>RKC PART_NAME NOT NULL</code>
KING.CODE_POSITION	KING.CODE_SALARY_BANK	KING.CODE_SERVER_COMPUTER	KING.CODE_SKILL_CATEGORY	KING.CODE_TOOL
<code>RKC POSITION_CODI NOT NULL</code>	<code>RKC SALARY_BANK_CODI NOT NULL</code>	<code>RKC SERVER COMPUTER_CODI NOT NULL</code>	<code>RKC SKILL_CATEGORY_CODI NOT NULL</code>	<code>RKC TOOL_CODI NOT NULL</code>
<code>RKC POSITION_NAME NOT NULL</code>	<code>RKC SALARY_BANK_NAME NOT NULL</code>	<code>RKC SERVER COMPUTER_NAME NOT NULL</code>	<code>RKC SKILL_CATEGORY_NAME NOT NULL</code>	<code>RKC TOOL_NAME NOT NULL</code>
KING.CODE_DOCUMENT_TYPE				
<code>RKC DOCUMENT_TYPE_CODI NOT NULL</code>				
<code>RKC DOCUMENT_TYPE_NAME</code>				
<code>RKC DOCUMENT_TYPE_FORM_CONTENT</code>				
<code>RKC DOCUMENT_TYPE_FORM_TITLE</code>				

DB테이블 설계 • 참조

프리랜서 최종 학력 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_HIGHESTLEVEL_EDU	HIGHESTLEVEL_EDU_CODE	최종 학력	최종 학력 번호	Y			VARCHAR2(10)		
	HIGHESTLEVEL_EDU_NAME		최종 학력 목록				VARCHAR2(30)		

프리랜서 현재 직업 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_JOB	JOB_CODE	현재 직업	현재 직업 번호	Y			VARCHAR2(10)		
	JOB_NAME		현재 직업 목록				VARCHAR2(30)		

프리랜서 파견 역할 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_DISPATCH_ROLE	DISPATCH_ROLE_CODE	파견 역할	파견 역할 번호	Y			VARCHAR2(10)		
	DISPATCH_ROLE_NAME		파견 역할 목록				VARCHAR2(30)		

프리랜서 근무 가능 지역 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_LOCATION	LOCATION_CODE	근무 가능 지역	근무 가능 지역 번호	Y			VARCHAR2(10)		
	LOCATION_NAME		근무 가능 지역 목록				VARCHAR2(30)		

DB테이블 설계•참조

프리랜서 선호분야 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_FAVORITE_PART	FAVORITE_PART_CODE	선호분야 이름	최종 학력 번호	Y			VARCHAR2(10)		
	FAVORITE_PART_NAME		최종 학력 목록				VARCHAR2(30)		

프리랜서 스킬분야 목록

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_SKILL_CATEGORY	SKILL_CATEGORY_CODE	스킬분야	스킬분야 번호	Y			VARCHAR2(10)		
	SKILL_CATEGORY_NAME		스킬분야 목록				VARCHAR2(30)		

프리랜서 블랙개발자 이유

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_BLACK_REASON	BLACK_REASON_CODE	블랙개발자 이유	스킬분야 번호	Y			VARCHAR2(10)		
	BLACK_REASON_NAME		블랙 개발자 이유 목록				VARCHAR2(30)		

DB테이블 설계 참조

정규직 병역 여부

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_GO_ARMY	GO_ARMY_CODE	병역 이름	병역 코드	Y			VARCHAR2(10)		
	BLACK_REASON_NAME		병역 여부 이름				VARCHAR2(30)		

급여통장번호 / 은행명

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_SALARY_BANK	SALARY_BANK_CODE	은행 이름	은행 코드	Y			VARCHAR2(10)		
	SALARY_BANK_NAME		은행 이름				VARCHAR2(30)		

학교 학부 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_HAKBU	HAKBU_CODE	학부 이름	학부 코드	Y			VARCHAR2(10)		
	HAKBU_NAME		학부 이름				VARCHAR2(30)		

정규직 회사경력 임무 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_DUTY	DUTY_CODE	업무 이름	업무 코드	Y			VARCHAR2(10)		
	DUTY_NAME		업무 이름				VARCHAR2(30)		

DB테이블 설계 참조

정규직 회사경력 분야 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_PART	PART_CODE	분야 이름	분야 코드	Y			VARCHAR2(10)		
	PART_NAME		분야 이름				VARCHAR2(30)		

정규직 회사경력 부서 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_DEP	DEP_CODE	부서 이름	부서 코드	Y			VARCHAR2(10)		
	DEP_NAME		부서 이름				VARCHAR2(30)		

정규직 회사경력 직위 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_POSITION	POSITION_CODE	직위 이름	직위 코드	Y			VARCHAR2(10)		
	POSITION_NAME		직위 이름				VARCHAR2(30)		

정규직 회사경력 고용형태 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_HIRE_TYPE	HIRE_TYPE_CODE	고용형태 이름	고용형태 코드	Y			VARCHAR2(10)		
	HIRE_TYPE_NAME		고용형태 이름				VARCHAR2(30)		

DB테이블 설계 참조

자격증 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_LICENSE	LICENSE_CODE	자격증 이름	자격증 코드	Y			VARCHAR2(10)		
	LICENSE_NAME		자격증 이름				VARCHAR2(30)		

외국어 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_FOREIGN_LANGUAGE	FOREIGN_LANGUAGE_CODE	외국어 이름	외국어 코드	Y			VARCHAR2(10)		
	FOREIGN_LANGUAGE_NAME		외국어 이름				VARCHAR2(30)		

정규직 관심분야 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_INTEREST	INTEREST_CODE	관심분야 이름	관심분야 코드	Y			VARCHAR2(10)		
	INTEREST_NAME		관심분야 이름				VARCHAR2(30)		

정규직 개발기종 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_SERVER_COMPUTER	SERVER COMPUTER_CODE	기종 이름	기종 코드	Y			VARCHAR2(10)		
	SERVER COMPUTER_NAME		기종 이름				VARCHAR2(30)		

DB테이블 설계 참조

정규직 개발경력 OS 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_OS	OS_CODE OS_NAME	OS 이름	OS 코드 OS 이름	Y			VARCHAR2(10) VARCHAR2(30)		

정규직 개발경력 언어 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_IT_LANGUAGE	IT_LANGUAGE_CODE IT_LANGUAGE_NAME	언어 이름	언어 코드 언어 이름	Y			VARCHAR2(10) VARCHAR2(30)		

정규직 개발경력 DBMS 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_DBMS	DBMS_CODE DBMS_NAME	DBMS 이름	DBMS 코드 DBMS 이름	Y			VARCHAR2(10) VARCHAR2(30)		

정규직 개발경력 TOOL 이름

물리적테이블명	물리적 컬럼명	논리적테이블명	논리적컬럼명	PK	FK	NULL	TYPE	DEFAULT	참조 테이블
CODE_TOOL	TOOL_CODE TOOL_NAME	TOOL 이름	TOOL 코드 TOOL 이름	Y			VARCHAR2(10) VARCHAR2(30)		

04

Class / JSP / Query 설계

CLASS 설계



정규직
개발자



프리랜서
개발자

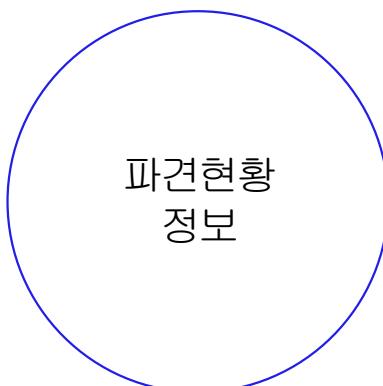
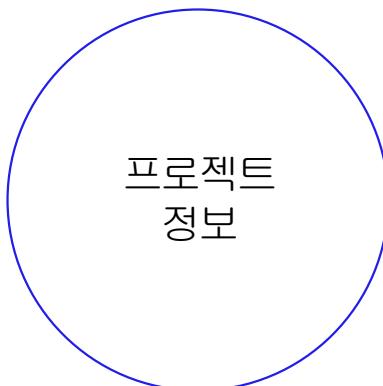
Dev Class (8개)

Class	Description
DevController	가상주소 /DevForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
DevService	DevServiceImpl에서 사용할 메소드 선언
DevServiceImp	정규직 정보의 입력/수정/삭제에 관한 DevService의 메소드를 오버라이딩
DevDTO	1개의 정규직정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언
DevDAO	정규직 정보의 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
DevSearchDTO	정규직페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언
DevSearchService	DevSearchImpl에서 사용할 메소드 선언
DevSearchServiceImp	정규직페이지의 검색에 관한 DevSearchService 의 메소드를 오버라이딩

Freelancer Class (8개)

Class	Description
FreelancerController	가상주소 /FreelancerForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
FreelancerService	FreelancerServiceImpl에서 사용할 메소드 선언
FreelancerServiceImp	프리랜서 정보의 입력/수정/삭제에 관한 FreelancerService의 메소드를 오버라이딩
FreelancerDTO	1개의 프리랜서정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언
FreelancerDAO	프리랜서 정보의 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
FreelancerSearchDTO	프리랜서페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언
FreelancerSearchService	FreelancerSearchImpl에서 사용할 메소드 선언
FreelancerSearchServiceImp	프리랜서페이지의 검색에 관한 FreelancerSearchService 의 메소드를 오버라이딩

CLASS 설계



Prj Class (8개)

Class	Description
PrjController	가상주소 /PrjForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
PrjService	PrjServiceImpl에서 사용할 메소드 선언
PrjServiceImp	프로젝트 정보의 입력/수정/삭제에 관한 PrjService의 메소드를 오버라이딩
PrjDTO	1개의 프로젝트정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언
PrjDAO	프로젝트 정보의 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
PrjSearchDTO	프로젝트페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언
PrjSearchService	PrjSearchImpl에서 사용할 메소드 선언
PrjSearchServiceImp	프로젝트페이지의 검색에 관한 PrjSearchService 의 메소드를 오버라이딩

Dispatch Class (8개)

Class	Description
DispatchController	가상주소 /DispatchForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
DispatchService	DispatchServiceImpl에서 사용할 메소드 선언
DispatchServiceImp	파견현황 정보의 입력/수정/삭제에 관한 DispatchService의 메소드를 오버라이딩
DispatchDTO	1개의 파견정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언
DispatchDAO	파견 정보의 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
DispatchSearchDTO	파견페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언
DispatchSearchService	DispatchSearchImpl에서 사용할 메소드 선언
DispatchSearchServiceImp	파견페이지의 검색에 관한 DispatchSearchService 의 메소드를 오버라이딩

CLASS 설계

결재현황
정보

Gyeoljae Class (6개)

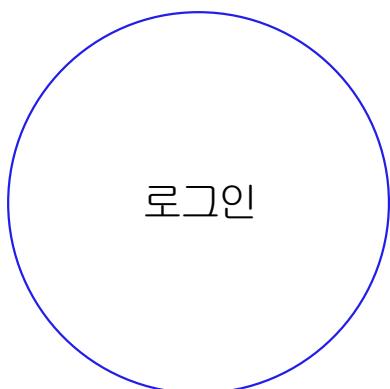
Class	Description
GyeoljaeController	가상주소 /GyeoljaeForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
GyeoljaeService	GyeoljaeServiceImp에서 사용할 메소드 선언
GyeoljaeServiceImp	결재 정보의 입력/수정/삭제/검색에 관한 GyeoljaeService의 메소드를 오버라이딩
GyeoljaeDTO	1개의 결재정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언
GyeoljaeDAO	결재 정보의 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
GyeoljaeSearchDTO	검색페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언

공지사항

Board Class (6개)

Class	Description
BoardController	가상주소 /BoardForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 JSP 페이지명과 응답물을 저장하고 리턴
BoardService	BoardServiceImp에서 사용할 메소드 선언
BoardServiceImp	공지사항 정보의 입력/수정/삭제에 관한 BoardService의 메소드를 오버라이딩
BoardDTO	1개의 공지사항의 상세정보 데이터를 담아 이동하는 getter/setter 선언
BoardDAO	공지사항 정보를 입력/수정/삭제/상세정보의 적용행의 개수를 리턴하는 메소드 선언
BoardSearchDTO	공지사항페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언

CLASS 설계



Login Class (2개)

Class	Description
LoginController	가상주소 /LoginForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 jsp 페이지명과 응답물을 저장하고 리턴
LoginDAO	DB에 저장된 로그인 계정 정보와 일치하는 행의 개수를 리턴하는 메소드 선언

Util Class

Class	Description
Util.java	검색 화면에 필요한 페이징 처리 관련 데이터를 HashMap 객체에 저장해 리턴하는 메소드 선언

JSP 설계*

JSP PAGE MAP (1/2)

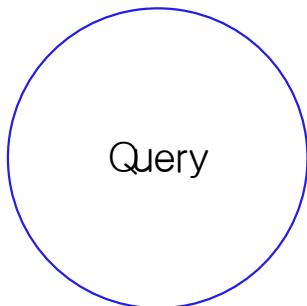
JSP PAGE	Description
common.jsp	JSP 페이지에서 공용으로 사용할 태그를 모아둔 JSP 페이지
boardList.jsp	공지사항을 보여주는 페이지
boardRegForm.jsp	공지사항을 작성 할 수 있는 JSP 페이지
boardUpDelForm.jsp	공지사항 수정 삭제를 할 수 있는 JSP 페이지
boardDetailFrom.jsp	공지사항 상세화면을 보여주는 JSP 페이지
loginForm.jsp	로그인 화면을 보여주는 JSP 페이지
mainpage.jsp	메인 화면을 보여주는 JSP 페이지
registDev.jsp	정규직 개발자 등록화면을 보여주는 JSP 페이지
searchDev.jsp	정규직 개발자 검색화면을 보여주는 JSP 페이지
registFreeDev.jsp	프리랜서 개발자 등록 화면을 보여주는 JSP 페이지
searchFreeDev.jsp	프리랜서 개발자 검색화면을 보여주는 JSP 페이지
registPrj.jsp	프로젝트 등록 화면을 보여주는 JSP 페이지
searchPrj.jsp	프로젝트 검색화면을 보여주는 JSP 페이지

JSP 설계

JSP PAGE MAP (2/2)

JSP PAGE	Description
gyeoljaeList.jsp	결재 현황을 보여주는 JSP페이지
gyeoljaeRegForm.jsp	결재사항을 등록하는 JSP 페이지
gyeoljaeDetailForm.jsp	결재 사항의 상세정보를 볼 수 있는 JSP 페이지
gyeoljaeUpDelForm.jsp	결재 사항의 수정 삭제를 할 수 있는 JSP 페이지
boardDetailFrom.jsp	공지사항 상세화면을 보여주는 JSP 페이지
error.jsp	관리자 계정이 아닌 일반 계정으로 관리자에게 허락된 기능을 시도 할 때 보여지는 JSP 페이지

QUERY 설계



.Xml(8개)

Class	Description
Mapper_board.Xml	공지사항의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언
Mapper_dev.Xml	정규직개발자의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언
Mapper_dispatch.Xml	파견현황의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언
Mapper_freelancer.Xml	프리랜서의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언
Mapper_gyeoljae.Xml	결재현황의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언
Mapper_main.Xml	메인페이지의 상세정보를 구현하는 쿼리문 선언
Mapper_login.Xml	로그인페이지의 로그인을 구현하는 쿼리문 선언
Mapper_prj.Xml	프로젝트의 입력/수정/삭제/상세정보를 구현하는 쿼리문 선언

04

Method 설계

Method 설계 • 로그인

클래스명	메소드	기능
LoginController	ModelAndView loginForm(HttpServletRequest session)	가상주소 /loginForm.do 로 접근하면 호출되는 메소드 선언
	Int loginProc (@RequestParam(value="mid") String mid ,@RequestParam(value="pwd") String pwd ,@RequestParam(value="autologin", required=false) String autologin ,HttpServletRequest session, HttpServletResponse response)	가상주소 /loginProc.do 로 접근하면 호출되는 메소드 선언
LoginDAO	LoginDAOImp 클래스의 메소드와 동일	인터페이스로 LoginDAOImp 의 메소드 목록을 규정
LoginDAOImp	Int getLoginIdCnt(Map<String, String> idPwd)	아이디와 비밀번호를 리턴하는 메소드 선언

Method 설계 • 정규직

클래스명	메소드	기능
DevController	ModelAndView searchDev(DevSearchDTO devSearchDTO, HttpSession session)	가상주소 /searchDev.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> searchDevDetail(DevSearchDTO devSearchDTO ,@RequestParam(value="dev_uid") String dev_uid)	가상주소 /searchDevDetail.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getDevListDetailMap(DevSearchDTO devSearchDTO)	[정규직 개발자 상세보기 정보]를 가져오는 메소드 선언
	int deleteDevInfo(DevDTO devDTO,@RequestParam(value="del_dev_uid") int del_dev_uid) throws Exception	가상주소 /deleteDevInfo.do 로 접근하면 호출하는 메소드 선언
	int updateDevInfo(DevDTO devDTO,@RequestParam(value="del_dev_uid") int del_dev_uid) throws Exception	가상주소 /updateDevInfo.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getDevListMap(DevSearchDTO devSearchDTO)	[정규직 개발자 정보]를 가져오는 메소드 선언
	Map<String, String> DevRegProc(DevDTO devDTO)	가상주소 /registDevProc.do 로 접근하면 호출하는 메소드 선언
DevDTO	getter/setter 메소드로 구성	[정규직 개발자 정보]의 입력/수정 관련 데이터를 담아 이동하는 getter/setter 메소드 선언
DevDAO	DevService 클래스의 메소드와 동일	인터페이스로 DevService 의 메소드 목록을 규정
DevService	DevServiceImp 클래스의 메소드와 동일	인터페이스로 DevServiceImp의 메소드 목록을 규정
DevServiceImp	Int insertDevInfo(DevDTO devDTO) throws Exception	[정규직 개발자 기본정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int dev_emergency_line(DevDTO devDTO) throws Exception	[정규직 개발자 긴급연락망 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int highSchool_info(DevDTO devDTO) throws Exception;	[정규직 개발자 고등학교 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int juniorCollege_info(DevDTO devDTO) throws Exception;	[정규직 개발자 전문대 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언

Method 설계 • 정규직

클래스명	메소드	기능
DevServiceImp	int university_info(DevDTO devDTO) throws Exception	[정규직 개발자 대학교 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int master_info(DevDTO devDTO) throws Exception	[정규직 개발자 대학교 석사 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int doctor_info(DevDTO devDTO) throws Exception	[정규직 개발자 대학교 박사 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_family_info(DevDTO devDTO) throws Exception	[정규직 개발자 가족관계 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int past_company_info(DevDTO devDTO) throws Exception	[정규직 개발자 회사 경력 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_license(DevDTO devDTO) throws Exception	[정규직 개발자 자격증 정보 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int dev_reading_language_info(DevDTO devDTO) throws Exception	[정규직 개발자 독해 외국어 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int devSpeaking_language_info(DevDTO devDTO) throws Exception	[정규직 개발자 회화 외국어 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int itCenter_info(DevDTO devDTO) throws Exception	[정규직 개발자 IT 교육 센터 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_liking_info(DevDTO devDTO) throws Exception	[정규직 개발자 기호 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_interest_info(DevDTO devDTO) throws Exception	[정규직 개발자 관심분야 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_past_prj_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_server_computer_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 기종 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언
	int dev_OS_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 OS 정보 입력 후 입력 적용 행의 개수]리턴하는 메소드 선언

Method 설계 • 정규직

클래스명	메소드	기능
DevServiceImp	int dev_it_language_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 언어 정보 입력 후 입력 적용 행의 개수]를 리턴하는 메소드 선언
	int dev_DBMS_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 DBMS 정보 입력 후 입력 적용 행의 개수]를 리턴하는 메소드 선언
	int dev_tool_info(DevDTO devDTO) throws Exception	[정규직 개발자 개발 경력 TOOL 정보 입력 후 입력 적용 행의 개수]를 리턴하는 메소드 선언
	int deleteDevInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_emergency_line(DevDTO devDTO) throws Exception	[1개 정규직 개발자 긴급연락망 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_highSchool_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 고등학교 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_juniorCollege_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 전문대 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_university_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_master_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 석사 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_doctor_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 박사 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_past_company_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 회사경력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_family_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 가족관계 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_license(DevDTO devDTO) throws Exception	[1개 정규직 개발자 자격증 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_reading_language_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 외국어 독해 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언

Method 설계•정규직

클래스명	메소드	기능
DevServiceImp	int delete_dev_speaking_language_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 외국어 회화 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_itCenter_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 IT교육센터 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_liking_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 기호 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_interest_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 관심분야 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_past_prj_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_server_computer_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 기종 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_OS_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 OS 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_it_language_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 언어 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_DBMS_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 DBMS 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_dev_tool_info(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 TOOL 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevEmergencyInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 긴급연락망 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevLikingInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 기호 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevPastPrjInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 개발경력 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언

Method 설계 • 정규직

클래스명	메소드	기능
DevServiceImp	int updateDevItcenterInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 IT교육센터 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevPastCompanyInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 회사경력 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevHighschoolInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 고등학교 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevjuniorcollegeInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 전문대 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevUniversityInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevMasterInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 석사 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevDoctorInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 대학교 박사 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateDevFamilyInfo(DevDTO devDTO) throws Exception	[1개 정규직 개발자 가족관계 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
DevSearchService	DevSearchServiceImp 클래스의 메소드와 동일	인터페이스로 DevSearchServiceImp 의 메소드 목록을 규정
DevSearchServiceImp	DevDTO getDevDetail(String dev_uid) throws Exception	[1개의 정규직 개발자 정보]의 번호를 리턴하는 메소드 선언

Method 설계 • 프리랜서

클래스명	메소드	기능
FreelancerController	ModelAndView searchFreeDev(FreelancerSearchDTO freelancerSearchDTO,HttpSession session)	가상주소 /searchFreeDev.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> searchFreeDevDetail(FreelancerSearchDTO freelancerSearchDTO,@RequestParam(value="free_uid") String free_uid)	가상주소 /searchFreeDevDetail.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getFreelancerListDetailMap(FreelancerSearchDTO freelancerSearchDTO)	[프리랜서 개발자 상세정보]를 가져오는 메소드 선언
	int deleteFreeInfo(FreelancerDTO freelancerDTO,@RequestParam(value="del_free_uid") int del_free_uid) throws Exception	가상주소 /deleteFreeInfo.do 로 접근하면 호출하는 메소드 선언
	int_updateFreeInfo(FreelancerDTO freelancerDTO,@RequestParam(value="del_free_uid") int del_free_uid) throws Exception	가상주소 /updateFreeInfo.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getFreelancerListMap(FreelancerSearchDTO freelancerSearchDTO)	[프리랜서 개발자 정보]를 가져오는 메소드 선언
	Map<String, String> freelancerRegProc(FreelancerDTO freelancerDTO)	가상주소 /registFreeDevProc.do 로 접근하면 호출하는 메소드 선언
FreelancerDTO	getter/setter 메소드로 구성	[프리랜서 개발자 정보]의 입력/수정 관련 데이터를 담아 이동하는 getter/setter 메소드 선언
FreelancerDAO	FreeService 클래스의 메소드와 동일	인터페이스로 FreeService 의 메소드 목록을 규정
FreelancerService	FreeServiceImp 클래스의 메소드와 동일	인터페이스로 FreeServiceImp의 메소드 목록을 규정
FreelancerServiceImp	int insertFreeInfo(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 기본정보 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int free_can_work_location_info(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 근무가능지역 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int free_favorite_part_code(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 선호분야 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int free_license(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 자격증 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언

Method 설계 • 프리랜서

클래스명	메소드	기능
FreelancerServiceImp	int free_skill_category(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 사용기술 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int free_year_info(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 개발년차 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int is_black_info(FreelancerDTO freelancerDTO) throws Exception	[프리랜서 개발자 블랙개발자 여부 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int deleteFreeInfo(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 기본 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_free_favorite_part(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 선호분야 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_free_license(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 자격증 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_free_skill_category(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 사용 스킬 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_free_year_info(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 개발년차 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_is_black_info(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 블랙개발자 여부 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int delete_free_can_work_location(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 근무가능지역 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int updateFreeInfo(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 정보 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int updateFreeYearInfo(FreelancerDTO freelancerDTO) throws Exception	[1개 프리랜서 개발자 ?? 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
FreelancerSearchService	DevSearchServiceImp 클래스의 메소드와 동일	인터페이스로 FreelancerSearchServiceImp 의 메소드 목록을 규정
DevSearchServiceImp	FreelancerDTO getFreeDetail(String free_uid) throws Exception	[1개의 프리랜서 개발자 정보]의 번호를 리턴하는 메소드 선언

Method 설계 • 프로젝트

클래스명	메소드	기능
PrjController	ModelAndView searchPrj(PrjSearchDTO prjSearchDTO, HttpSession session)	가상주소 /searchPrj.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> searchPrjDetail(PrjSearchDTO prjSearchDTO, @RequestParam(value="prj_uid") String prj_uid)	가상주소 /searchPrjDetail.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getPrjListMap(PrjSearchDTO prjSearchDTO)	[프로젝트 상세정보]를 가져오는 메소드 선언
	int deletePrjDetail(PrjDTO prjDTO, @RequestParam(value="del_prj_uid") String del_prj_uid) throws Exception	가상주소 /deletePrjInfo.do 로 접근하면 호출하는 메소드 선언
	updatePrjDetail(PrjDTO prjDTO, @RequestParam(value="del_prj_uid") String del_prj_uid) throws Exception	가상주소 /updatePrjInfo.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getPrjListDetailMap(PrjSearchDTO prjSearchDTO)	[프로젝트 정보]를 가져오는 메소드 선언
	Map<String, String> PrjRegProc(PrjDTO prjDTO)	가상주소 /registPrjProc.do 로 접근하면 호출하는 메소드 선언
PrjDTO	getter/setter 메소드로 구성	[프로젝트 정보]의 입력/수정 관련 데이터를 담아 이동하는 getter/setter 메소드 선언
PrjDAO	PrjService 클래스의 메소드와 동일	인터페이스로 PrjService 의 메소드 목록을 규정
PrjService	PrjServiceImp 클래스의 메소드와 동일	인터페이스로 PrjServiceImp의 메소드 목록을 규정
PrjServiceImp	int insertPrjInfo(PrjDTO prjDTO) throws Exception;	[프로젝트 기본정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_dev_person_info(PrjDTO prjDTO) throws Exception;	[프로젝트 개발인력 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_dev_license_info(PrjDTO prjDTO) throws Exception;	[프로젝트 개발인력의 자격증 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_oper_person_info(PrjDTO prjDTO) throws Exception;	[프로젝트 운영인력 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언

Method 설계•프로젝트

클래스명	메소드	기능
PrjServiceImp	int prj_oper_license_info(PrjDTO prjDTO) throws Exception	[프로젝트 운영인력의 자격증 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_publ_person_info(PrjDTO prjDTO) throws Exception	[프로젝트 퍼블리셔인력 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_publ_license_info(PrjDTO prjDTO) throws Exception	[프로젝트 퍼블리셔인력의 자격증 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_desi_person_info(PrjDTO prjDTO) throws Exception	[프로젝트 디자인인력 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int prj_desi_license_info(PrjDTO prjDTO) throws Exception	[프로젝트 디자인인력의 자격증 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int deletePrj(PrjDTO prjDTO) throws Exception	[프로젝트 기본정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deletePrjJoin(PrjDTO prjDTO) throws Exception	[파견인원을 포함한 프로젝트 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteDev(PrjDTO prjDTO) throws Exception	[프로젝트 개발인력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteDevLicens(PrjDTO prjDTO) throws Exception	[프로젝트 개발인력 자격증 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteOper(PrjDTO prjDTO) throws Exception	[프로젝트 운영인력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteOperLicens(PrjDTO prjDTO) throws Exception	[프로젝트 운영인력 자격증 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deletePubl(PrjDTO prjDTO) throws Exception	[프로젝트 퍼블리셔인력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deletePublLicens(PrjDTO prjDTO) throws Exception	[프로젝트 퍼블리셔인력 자격증 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteDesi(PrjDTO prjDTO) throws Exception	[프로젝트 디자인인력 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int deleteDesiLicens(PrjDTO prjDTO) throws Exception	[프로젝트 디자인인력 자격증 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int updatePrjDetail(PrjDTO prjDTO)	[프로젝트 정보 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언

Method 설계 • 파견

클래스명	메소드	기능
DispatchController	ModelAndView searchDispatch(DispatchSearchDTO dispatchSearchDTO,PrjSearchDTO prjSearchDTO,HttpSession session,@RequestParam(value="prj_num", required=false) String prj_num)	가상주소 /searchDispatch.do 로 접근하면 호출하는 메소드 선언
	Map<String, Object> getDispatchListMap(DispatchSearchDTO dispatchSearchDTO)	[파견현황 정보]를 가져오는 메소드 선언
	Map<String, Object> getAllDevListMap(String prj_num)	[프리랜서 개발자 정규직 개발자 정보] 전부를 가져오는 메소드 선언
	Map<String, Object> getPrjListMap(PrjSearchDTO prjSearchDTO)	[파견 현황 정보] 검색에 필요한 정보를 가져오는 메소드 선언
	public int insertDispatch(DispatchSearchDTO dispatchSearchDTO)	가상주소 /insertDispatch.do 로 접근하면 호출하는 메소드 선언
	public int deleteDispatch(DispatchSearchDTO dispatchSearchDTO)	가상주소 /deleteDispatch.do 로 접근하면 호출하는 메소드 선언
DispatchDTO	getter/setter 메소드로 구성	[파견현황 정보]의 입력에 관련 데이터를 담아 이동하는 Getter/setter 메소드 선언
DispatchSearchDTO	getter/setter 메소드로 구성	[파견현황 정보]의 검색에 관련 데이터를 담아 이동하는 Getter/setter 메소드 선언
DispatchService	DispatchServiceImp 클래스의 메소드와 동일	인터페이스로 DispatchServiceImp의 메소드 목록을 규정
DispatchServiceImp	int insertDispatch(DispatchSearchDTO dispatchSearchDTO) throws Exception	[파견 정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int deleteDispatch(DispatchSearchDTO dispatchSearchDTO)	[파견 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
DispatchSearchService	DispatchSearchServiceImp 클래스의 메소드와 동일	인터페이스로 DispatchSearchServiceImp의 메소드 목록을 규정
DispatchSearchServiceImp	DispatchDTO getDispatch(int d_no)	[1개의 파견]을 리턴하는 메소드 선언

Method 설계•결제

클래스명	메소드	기능
GyeoljaeController	public ModelAndView gyeoljaeDetailForm(@RequestParam(value="gyeoljae_num") String gyeoljae_num)	가상주소 /gyeoljaeDetailForm.do로 접근하면 호출하는 메소드 선언
	Map<String, String> gyeoljaeResult(GyeoljaeDTO gyeoljaeDTO)	가상주소 /gyeoljaeResult.do로 접근하면 호출하는 메소드 선언
	ModelAndView gyeoljaeUpDelForm(@RequestParam(value="gyeoljae_num") String gyeoljae_num)	가상주소 /gyeoljaeUpDelForm.do로 접근하면 호출하는 메소드 선언
	Map<String, String> gyeoljaeUpProc(GyeoljaeDTO gyeoljaeDTO)	가상주소 /gyeoljaeUpProc.do로 접근하면 호출하는 메소드 선언
	int gyeoljaeDelProc(GyeoljaeDTO gyeoljaeDTO)	가상주소 /gyeoljaeDelProc.do로 접근하면 호출하는 메소드 선언
	ModelAndView mainGyeoljaeList(GyeoljaeSearchDTO gyeoljaeSearchDTO,HttpSession session)	가상주소 /gyeoljaeList.do로 접근하면 호출하는 메소드 선언
	Map<String, Object> getGyeoljaeSearchResultMap(GyeoljaeSearchDTO gyeoljaeSearchDTO)	결재 검색 결과 정보를 구하고 Map<String, Object> 객체에 담아 리턴하는 메소드 선언
GyeoljaeDTO	Map<String, String> gyeoljaeRegProc(GyeoljaeDTO gyeoljaeDTO)	가상주소 /gyeoljaeRegProc.do로 접근하면 호출하는 메소드 선언
GyeoljaeSearchDTO	getter/setter 메소드로 구성	[결재 정보]의 입력/수정에 관련 데이터를 담아 이동하는 Getter/setter 메소드 선언
GyeoljaeService	getter/setter 메소드로 구성	[결재 정보]의 검색에 관련 데이터를 담아 이동하는 Getter/setter 메소드 선언
GyeoljaeServiceImp	GyeoljaeServiceImp 클래스의 메소드와 동일	인터페이스로 GyeoljaeServiceImp의 메소드 목록을 규정
	GyeoljaeDTO getGyeoljae(String gyeoljae_num);	[매개변수에 해당하는 결재 정보]를 가져오는 메소드 선언
	int updateGyeoljae(GyeoljaeDTO gyeoljaeDTO) throws Exception	[1개 결재사항 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언
	int deleteGyeoljae(GyeoljaeDTO gyeoljaeDTO)	[1개 결재사항 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언
	int insertGyeoljae(GyeoljaeDTO gyeoljaeDTO) throws Exception;	[결재 사항정보를 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언
	int updateProgress(GyeoljaeDTO gyeoljaeDTO) throws Exception;	[결재 상태 정보를 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언

Method 설계 • 공지사항

클래스명	메소드	기능
BoardController	<code>ModelAndView boardList(BoardSearchDTO boardSearchDTO, HttpSession session)</code>	가상주소 /boardList.do 로 접근하면 호출되는 메소드 선언
	<code>Map<String, Object> getBoardSearchResultMap(BoardSearchDTO boardSearchDTO)</code>	게시판 검색 결과 정보들을 구해서, Map<String, Object> 객체에 담아 리턴하는 메소드 선언
	<code>ModelAndView boardDetailForm(@RequestParam(value="b_no") int b_no)</code>	가상주소 /boardDetailForm.do 로 접근하면 호출되는 메소드 선언
	<code>ModelAndView boardUpDelForm(@RequestParam(value="b_no") int b_no)</code>	가상주소 /boardUpDelForm.do 로 접근하면 호출되는 메소드 선언
	<code>Map<String, String> boardUpProc(@ValidBoardDTO boardDTO ,BindingResult bindingResult)</code>	가상주소 /boardUpProc.do 로 접근하면 호출되는 메소드 선언
	<code>boardDelProc(BoardDTO boardDTO)</code>	가상주소 /boardDelProc.do 로 접근하면 호출되는 메소드 선언
	<code>ModelAndView boardRegForm()</code>	가상주소 /boardRegForm.do 로 접근하면 호출되는 메소드 선언
boardDTO	<code>Map<String, String> boardRegProc(BoardDTO boardDTO)</code>	가상주소 /boardRegProc.do 로 접근하면 호출되는 메소드 선언
	getter/setter 메소드로 구성	[공지사항]의 입력/수정 관련 데이터를 담아 이동하는 getter/setter 메소드 선언
boardSearchDTO	getter/setter 메소드로 구성	[공지사항]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언
BoardService	BoardServiceImp 클래스의 메소드와 동일	인터페이스로 BoardServiceImp의 메소드 목록을 규정
BoardServiceImp	<code>BoardDTO getBoard(int b_no, boolean isBoardDetailForm)</code>	[1개의 게시판글] 리턴하는 메소드 선언
	<code>int updateBoard(BoardDTO boardDTO) throws Exception</code>	[1개 게시판] 수정 실행하고 수정 적용행의 개수를 리턴하는 메소드 선언
	<code>int deleteBoard(BoardDTO boardDTO)</code>	[1개 게시판] 삭제 실행하고 삭제 적용행의 개수를 리턴하는 메소드 선언
	<code>int insertBoard(BoardDTO boardDTO) throws Exception</code>	[게시판 글 입력 후 입력 적용행의 개수] 리턴하는 메소드 선언

05

상세화면

로그인 | 회원가입 | 공지사항
정규직, 프리랜서관리 | 프로젝트관리
파견관리 | 보고서결재

로그인



1 아이디

- 아이디 입력칸
- 공백없이 영소문자로 시작하는 숫자,_ 포함 4~20자 입력 가능

2 비밀번호

- 비밀번호 입력칸
- 공백없이 영소문자, 또는 숫자로 4~30자 입력 가능

3 자동 로그인

- 자동 로그인 체크 박스
- 자동 로그인 체크시 쿠키값 저장하여 아이디 비밀번호 자동입력

4 로그인 버튼

- 로그인 버튼 선택 시, DB에 저장된 계정정보 조회 후 일치하는 계정이 있을 경우, 메인 페이지로 이동
일치하는 계정이 없을 경우, 경고창 표시

메인화면• 01



- 1 **홈 버튼**
 - DEVGRU : 선택 시, 메인 페이지로 이동
- 2 **카테고리 버튼**
 - 선택 시, 각 카테고리별 검색 페이지로 이동

- 3 **로그아웃 버튼**
 - 로그아웃 버튼 클릭 시 기존 아이디에서 로그아웃하고 로그인 화면으로 전환
- 4 **더 보기 버튼**
 - 컨텐츠에 있는 더 보기 버튼 클릭 시 해당 페이지로 화면 전환

- 5 **통계**
 - 구글차트를 이용한 현황 가시화

메인화면• 02

공지사항

1
+

22	프로젝트 종료관련	권웅순	2023-11-09	15
21	신입 프리랜서 입사	권웅순	2023-11-07	30
20	신입 프리랜서 입사	권웅순	2023-11-07	22
19	신입 프리랜서 입사	권웅순	2023-11-07	8
18	신입 프리랜서 입사	권웅순	2023-11-07	5
17	신입 프리랜서 입사	권웅순	2023-11-07	1

결재 문서

1
+

2	반려	devgru 프로젝트에 관한 건	정선희 사장	2023-11-09 오후 05:33
	승인	근태 보고의 건	정선희 사장	2023-11-09 오후 05:11
	미결	CJ 차기 신소재 태크 개발에 관한 건	정선희 사장	2023-11-09 오후 05:08
	승인	연차 보고의 건	정선희 사장	2023-11-09 오후 05:05
	미결	지출 보고의 건	정선희 사장	2023-11-09 오후 05:02
	미결	근태 보고의 건	정선희 사장	2023-11-09 오후 04:49

DEVG RU INC

A 서울특별시 금천구 가산동 549-1 한라 원엔원 타워 지식산업센터
T 02-1004-1004 | F 02-1004-1004

1 더 보기

- 컨텐츠에 있는 더 보기 버튼 클릭 시 해당 페이지로 화면 전환

2 상세 보기

- 행 클릭 시 해당 글에 대한 상세보기 페이지로 화면 전환

프리랜서 정보 입력^{• 01}

프리랜서 기본 정보

1 이름 성별 남 여 이메일 핸드폰 전화번호 틀림

출생년도 년생 최종학력 출입년도 ---년 - 월

프리랜서 기본 나이

2 추정 개발년차 0년 0개월 (0) 나이
이력서 상 개발년차 년 개월 (연도-월-일 날짜 기준)

프리랜서 평가

3 블랙개발자 실력부족 파견지부적응 근태불성실 무단이탈 전화통화원치않음 기타
기타 평가
 최대 300자 입력

프리랜서 파견 정보

4 상담시각 연도-월-일 --::-- [지금] [비움]
현재직업 프리랜서 개발자 정규직 개발자 기타직업
스킬분야 JAVA>JSP .NET PHP ASP ABAP Delphi PB VC++ 웹디 기타
파견역할 개발 운영 분석설계 PL PM 웹디 publisher 기타
파견가능일 연도-월-일 확정
개발 단가 만원 ~ 만원 단가고정 협상후결정
운영 단가 만원 ~ 만원 단가고정 협상후결정

① 프리랜서 기본 정보 입력

- 이름 : 한글 또는 영어 2~15자 입력 가능
- 성별 : 남 또는 여 단일 선택
- 이메일 : 공백없이 @, ., com을 포함해 입력 가능
- 핸드폰 : 공백없이 010으로 시작하는 숫자 11자리 입력 가능
- 전화번호 틀림 : 체크박스 선택사항
- 출생년도 : 공백없이 숫자 4자리 입력 가능
- 최종학력 : 고졸, 전문대졸, 일반대학사졸, 일반대학석사졸, 일반대학박사졸 선택 가능
- 출입년도 : 선택 시, 년-월 시스템 캘린더 표시

② 프리랜서 기본 나이 입력

- 년 : 최대 0~20년 선택 가능
- 개월 : 최대 0~12개월 선택 가능
- 날짜 기준 : 선택 시, 년-월-일 시스템 캘린더 표시

③ 프리랜서 평가 입력

- 블랙개발자 체크박스 : 다중 선택 가능
- 기타 평가 : 공백포함 한영 최대 300자 입력 가능

④ 프리랜서 파견 정보 입력

- 상담시각 : 선택 시, 년-월-일-시간-분 시스템 캘린더 표시
[지금] : 선택 시, 현재 년-월-일-시간-분 표시
[비움] : 선택 시, 입력된 년-월-일-시간-분 초기화
- 현재직업 : 단일선택 라디오 버튼 제공
- 스킬분야 : 다중선택 체크박스 제공
- 파견역할 : 단일선택 라디오 버튼 제공
- 파견가능일 : 선택 시, 년-월-일 시스템 캘린더 표시
[확정] : 확정 여부 체크박스 제공
- 개발/운영단가 : 3자리 이상 숫자 입력 가능하며 최소단가는 최대단가 초과 불가
: 개발/운영단가 중 필수 택1
- [단가고정, 협상후결정] : 단일선택 체크박스 제공

프리랜서 정보 입력^{•02}

1

기사자격증 정보처리기사 정보처리산업기사
노트북 소유 노트북대여희망
근무가능지역 서울 인천 경기 강원 충북 충남 전북 전남 경북 경남 제주 국외
선호분야 교육 대부 대학 보험 쇼핑 언론 유통 은행 의료 전자/전기 정부기관 증권 카드 통신 기타
거주지 시(도)

2

기타요구
최대 300자 입력

전화상담시 주의사항
최대 300자 입력

3

① 프리랜서 파견정보 입력

- 기사자격증 : 체크박스 선택사항
- 노트북 소유 : 체크박스 선택사항
- 근무가능지역 : 체크박스 필수사항 및 다중선택 가능
- 선호분야 : 체크박스 필수사항 및 다중선택 가능
- 거주지 : 서울,인천,경기,강원,충북,전북,전남,경북,경남,국외에 한해 선택 제공

② 기타요구/ 전화상담시 주의사항

- 기타요구 : 공백포함 한영 최대 300자 입력 가능
- 전화상담시 주의사항 : 공백포함 한영 최대 300자 입력 가능

③ 취소/저장 버튼

- 취소 : 선택 시, 프리랜서 정보 검색 페이지로 이동
- 저장 : 선택 시, 유효성 검사 실행 후
 - 통과할 경우 : 프리랜서 정보 검색 페이지로 이동
 - 통과못할 경우 : 미통과 항목에 초점 제공 및 경고문구 표시

프리랜서 정보 입력 03

The form consists of several sections with numbered callouts:

- 1**: A large input field for general search.
- 2**: An 'or' operator dropdown.
- 3**: A search button.
- 4**: A 'Clear' button.
- 5**: A 'Search [Freelancer]' button.
- 6**: A section for 'White/Black developer' with checkboxes for 'White developer' and 'Black developer'.
- 7**: A section for 'Skills' with checkboxes for various programming languages.
- 8**: A section for 'Meeting date' with a date range selector and a 'Clear' button.
- 9**: A section for 'Virtual developer' with a date range selector and a 'Clear' button.
- 10**: A section for '자격증' (Qualification) with checkboxes for 'Information Processor' and 'Information Processing Industry Worker'.
- 11**: A section for '역할' (Role) with checkboxes for '개발' (Development), '운영' (Operation), '분석설계' (Analysis/Design), 'PL' (Project Leader), 'PM' (Project Manager), and '웹디' (Web Design).

① 키워드 검색

- 키워드 검색 입력칸 : 한영,숫자 30자에 한해 입력 가능

② or and

- 두가지 키워드 사이 or, and 구분하여 검색 가능하도록 옵션 제공

③ 검색 버튼

- 선택 시, 사용자 선택 조건에 한해 검색 결과 표시

④ 초기화 후 전부검색 버튼

- 선택 시, 사용자 선택 조건 초기화 후 모든 검색 결과 표시

⑤ [프리랜서] 입력 버튼

- 선택 시, 프리랜서 정보 입력 페이지로 이동

⑥ 화이트/블랙 개발자분

- 다중선택 체크박스 가능

⑦ 스킬분야

- 다중선택 체크박스 가능

⑧ 파견 가능일

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

⑨ 가상개발년차

- 최소, 최대 0~20년차 선택 가능

⑩ 자격증

- 다중선택 체크박스 가능

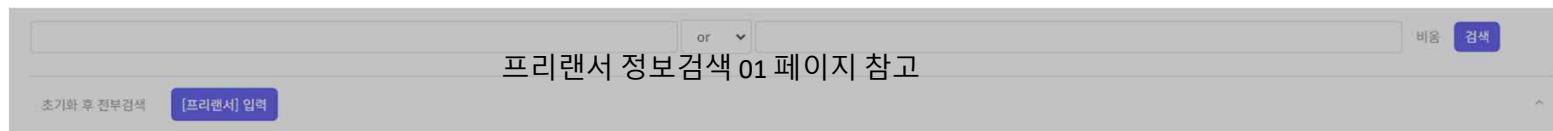
⑪ 역할

- 다중선택 체크박스 가능

⑫ 비움 버튼

- 선택 시, 해당 행에 입력된 값 모두 초기화

프리랜서 정보 검색[•] 01



①최종학력

- 다중선택 체크박스 가능

②전화상담일

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

③나이

- 최소, 최대 숫자 2자리 입력 가능

④성별

- 다중선택 체크박스 가능

⑤졸업년도

- 최소 1970년 ~ 최대 현재 년 선택 가능

⑥이력서확인일

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

⑦현재직업

- 다중선택 체크박스 가능

프리랜서 정보 검색^{•02}

1
검색개수 : 27 개
2
10
▼
행 보기

3
이름 오름차순
▼
나이 오름차순
▼
학력 내림차순
▼
가상개발년차 오름차순
▼
전화상담일 오름차순
▼
파견가능일 내림차순
▼
이력서확인일 내림차순
▼

4

번호	스킬분야	이름	성별	나이	학력	가상개발년차	파견가능일	파견가능일 확정	핸드폰	이메일	전화 상담일	이력서 확인일
27	NET, PHP, Delphi	고건린	남	26 살	일반대학졸	10 년	2023년 10월 11일	미확정	01076211883	iolv5210@naver.com	2023년 10월	2023년 10월 10일
26	Java/JSP, .NET, AB	권시린	여	26 살	일반대학졸	03 년	2023년 10월 11일	확정	01084143136	lrn8795@naver.com	2023년 10월	2023년 10월 10일
25	PHP, ABAP	김건우	남	23 살	고졸	06 년	2023년 10월 28일	확정	01059392890	oj4338@naver.com	2023년 10월	2024년 06월 10일
24	JAVA/JSP	김또깡	여	31 살	전문대졸	10 년	2023년 10월 11일	미확정	01011254334	imkk@nave.com	2023년 10월	2023년 10월 10일
23	Delphi, PB	김연향	남	26 살	일반대학졸	03 년	2023년 10월 11일	확정	01020465780	xm2171@naver.com	2023년 10월	2023년 10월 10일
22	JAVA/JSP	김정학	남	26 살	고졸	10 년	2023년 10월 11일	미확정	01046143386	gkr3386@gmail.com	2023년 10월	2023년 10월 10일
21	.NET, PHP	김주희	여	35 살	고졸	10 년	2023년 10월 11일	미확정	01002445834	oxw6780@naver.com	2023년 10월	2023년 10월 10일
20	JAVA/JSP	박나혁	남	47 살	고졸	10 년	2023년 10월 11일	미확정	01085689837	yop0674@naver.com	2023년 10월	2023년 10월 10일
19	PHP, ASP, ABAP	박재영	여	36 살	일반대학졸	10 년	2023년 11월 23일	확정	01022556754	df007@daum.net	2023년 11월	2023년 11월 07일
18	JAVA/JSP, 기타	서니	여	24 살	전문대졸	02 년	2026년 01월 01일	미확정	01030950055	ncoior17@gmail.com	2023년 10월	2023년 10월 10일

6

« < 1 [2] [3] > »

1 검색개수

- 검색 결과 개수 표시

2 N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

3 정렬기준

- 검색결과에 따른 정렬 기준 선택 제공

4 검색결과

- 번호, 스킬분야, 이름, 성별, 나이, 학력, 가상개발년차, 파견가능일, 파견가능일 확정여부, 핸드폰, 이메일, 전화상담일, 이력서 확인일 정보를 포함한 검색결과 기본 10개 표시

5 검색결과 상세

- 검색 결과 행 선택 시, 상세 정보 팝업 표시

6 페이지 이동

- 최소, 최대 페이지 바로가기 및 최대 페이지 표시. 번호 선택 시, 해당 페이지로 이동

프리랜서 정보 검색^{•03}

1

프리랜서 기본 정보

이름

성별 남 여

이메일

핸드폰

출생년도 년생

최종학력

졸업년도

프리랜서 기본 년차

추정 개발년차 0년 0개월 (6) 년차

이력서 상 개발년차 06 년 06 개월 (2023-07-08 날짜 기준)

프리랜서 평가

블랙개발자 실력부족 파견지부적응 근태불성실 무단이탈 전화통화원치않음 기타

기타 평가 근무 중 이탈하는 상황 있었음

프리랜서 파견 정보

상담시각 [지금] [비음]

현재직업 프리랜서 개발자 정규직 개발자 기타직업

스킬분야 JAVA>JSP .NET PHP ASP ABAP Delphi PB VC++ 웹디 기타

파견역할 개발 운영 분석설계 PL PM 웹디 publisher 기타

파견가능일 확정

개발 단가 만원 ~ 만원 단가고정 협상후결정

운영 단가 만원 ~ 만원 단가고정 협상후결정

기사자격증 정보처리기사 정보처리산업기사

노트북 소유 노트북대여회망

근무가능지역 서울 인천 경기 강원 충북 충남 전북 전남 경북 경남 제주 국외

선호지역 교육 대부 대학 보험 쇼핑 언론 유통 은행 의료 전자/전기 정부기관 증권 카드 통신 기타

거주지 시(도)

기타요구

전화상담시 주의사항

2

[닫기](#) [삭제](#) [수정](#)

3

4

1. 프리랜서 상세 정보 표시

- 검색결과에서 선택된 행에 상세정보를 팝업으로 표시 및 수정 비활성화

2. 닫기 버튼

선택 시, 팝업 닫힘

3. 삭제 버튼

- 선택 시, '정말 삭제하시겠습니까?' 알림 표시 후 확인 선택 시, '프리랜서 정보가 삭제되었습니다.' 표시 및 해당 정보 삭제되며 팝업 닫힘 변경된 프리랜서 검색 결과물은 팝업 닫힘과 함께 갱신되어 프리랜서 개발자 정보 검색 화면에 표시됨

4. 수정 버튼

- 선택 시, [완료] 버튼으로 문구 변경되며 이름 입력칸으로 초점 이동 및 모든 입력칸 수정 활성화

정규직 정보 입력• 01

1

아이디 암호 암호확인

이름 주민번호 핸드폰 (-없이 입력)

이메일 병역 (여성은 면제로 입력해 주세요.)

[주소] :

긴급연락처(사고 대비) [연락 받을 사람 이름] :

[관계] : (부모,형제,친척,지인)

[전화번호] :

2

급여통장번호 은행선택... (-없이 입력)

3

학력

학교명	학부	전공과목	부전공과목	졸업일
고등학교 졸				연도·월·일 <input type="text"/>
전문대 졸(대학 졸)				연도·월·일 <input type="text"/>
대학교 학사 졸 <input type="checkbox"/> 현입				연도·월·일 <input type="text"/>
대학교 석사 졸				연도·월·일 <input type="text"/>
대학교 박사 졸				연도·월·일 <input type="text"/>

4

1 정규직 기본 정보 입력

- 아이디 : 기존 DB에 등록되지 않은 영소문자로 시작하는 숫자_포함 4~20자 입력 가능
- 암호 : 영소문자 또는 숫자 4~20자 입력 가능
- 암호확인 : 기존 입력된 암호화 동일한 암호 입력 가능
- 이름 : 한글 또는 영어 2~15자 입력 가능
- 주민번호 : 앞자리 숫자 6자리 및 뒷자리 숫자 7자리 입력 가능
- 핸드폰 : 공백없이 010으로 시작하는 숫자 11자리 입력 가능
- 이메일 : 공백없이 @, ., com을 포함해 입력 가능
- 병역 : 군필,의가사제대,면제에 한해 선택 가능
- 주소 : 특수문자 및 기호를 포함한 한영 입력 가능
- 긴급연락처(이름) : 한영 2~15자 입력 가능
- 긴급연락처(관계) : 부모,형제,자식,친척,지인에 한해 선택 가능
- 긴급연락처(전화번호) : 공백없이 010으로 시작하는 숫자 11자리 입력 가능

2 급여통장번호

- 은행선택 : 하나은행,국민은행,카카오뱅크,신한은행에 한해 선택 가능
- 번호입력 : 숫자 14자리 입력 가능

3 학력

- 학력입력 : 학력 5학년 중 고등학교 필수입력
- 학교명 입력 시, 해당 행의 졸업일 필수 선택

4 학력 상세

- 고등학교 제외 입력 항목
- 고등학교 제외 학교 입력 시, 해당행의 학부,전공과목 필수입력

정규직 정보 입력 02

가족		기록 추가																										
관계	이름	직업(미성년자는 무직 입력)	출생년도	동거여부	형																							
<input type="button" value="▼"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="▼년"/>	<input type="checkbox"/> 동거	초기화																							
<input type="button" value="▼"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="▼년"/>	<input type="checkbox"/> 동거	행사제																							
<input type="button" value="▼"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="▼년"/>	<input type="checkbox"/> 동거	행사제																							
<input type="button" value="▼"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="▼년"/>	<input type="checkbox"/> 동거	행사제																							
<input type="button" value="▼"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="▼년"/>	<input type="checkbox"/> 동거	행사제																							
2 회사 경력 (개발 관련 최근 1건만 입력) <table border="1"> <tr> <td>회사명/업무</td> <td>분야</td> <td>부서</td> <td>직위</td> <td>고용형태</td> <td>연봉</td> <td>입사일/퇴사일</td> </tr> <tr> <td><input type="text"/> 회사명 <input type="button" value="▼"/></td> <td><input type="text"/></td> <td>입사일 : <input type="button" value="----년 --월"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>퇴사일 : <input type="button" value="----년 --월"/></td> </tr> </table>								회사명/업무	분야	부서	직위	고용형태	연봉	입사일/퇴사일	<input type="text"/> 회사명 <input type="button" value="▼"/>	<input type="text"/>	입사일 : <input type="button" value="----년 --월"/>							퇴사일 : <input type="button" value="----년 --월"/>				
회사명/업무	분야	부서	직위	고용형태	연봉	입사일/퇴사일																						
<input type="text"/> 회사명 <input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="text"/>	입사일 : <input type="button" value="----년 --월"/>																						
						퇴사일 : <input type="button" value="----년 --월"/>																						
3 자격증 <p><input type="checkbox"/> 정보처리기사 <input type="checkbox"/> 정보처리산업기사</p>																												

1 가족관계

- 관계 : 부,모,자녀,형제자매,배우자 외 한해 선택 가능
- 이름 : 한글 또는 영어 2~15자 입력 가능
- 직업 : 한글 또는 영어 2~20자 입력 가능
- 출생년도 : 최소 1910년 ~ 최대 현재년도 선택 가능
- 동거여부 : 체크박스 선택사항

* 가족추가 클릭 시 행 추가 / 행 삭제 클릭 시 행 삭제
 * 가족관계 선택 시, 동일 행 입력항목 필수 입력
 * 초기화 버튼 클릭 시 기입했던 내용들 초기화

2 회사경력

- 회사명 : 한글 또는 영어 2~30자 입력 가능
- 업무 : 개발,운영,분석설계,PL,PM,웹디에 대해서 선택 가능
- 분야 : 대부,대학,보험,쇼핑,언론 외 다수 항목에 대해서 선택 가능
- 부서 : 개발부,운영부,사업부에 대해서 선택 가능
- 직위 : 사원,주임,대리,과장 외 다수 항목에 대해서 선택 가능
- 고용형태 : 정규직,비정규직,인턴에 대해서 선택 가능
- 연봉 : 숫자 20자 입력 가능
- 입사일/퇴사일 : 선택 시 시스템 캘린더 표시 및
입사일 이전 퇴사일 혹은 퇴사일 이후 입사일 선택 불가

* 회사명 입력 시, 동일 행 입력항목 필수 입력

3 자격증

- 체크박스 선택사항

정규직 정보 입력 03

1 IT 교육센터

2 기호

3 개발경력

4 취소 저장

1 IT 교육센터

- 교육센터명 : 한영 및 숫자 2~30자 입력 가능
- 교육기간 : 최소 1개월 ~ 최대 12개월 선택 가능
- 교육수료일 : 선택 시 시스템 캘린더 표시
- 수료과목 : 한영 및 숫자 2~30자 입력 가능

* 교육센터명 입력 시, 동일 행 입력항목 필수 입력

2 기호

- 하루 흡연량 : 최소 0 ~ 최대 3 선택 가능
- 소주 주량 : 최소 0 ~ 최대 3 선택 가능
- 맥주 주량 : 최소 0 ~ 최대 3 선택 가능
- 관심분야 : 체크박스 다중선택 가능
1개 선택 필수

3 개발경력

- 프로젝트명 : 한영 및 숫자 2~30자 입력 가능
- 시작/종료일 : 선택 시 시스템 캘린더 표시 및 시작일 이전 종료일 혹은 종료일 이후 시작일 선택 불가
- 종료일 : 선택 시 시스템 캘린더 표시 및 종료일 이전 시작일 혹은 시작일 이후 종료일 선택 불가
- 파견단계 : 파견확정,파견중,파견종료에 한해서 선택 가능
- 고객사 : 한영 및 숫자 2~30자 입력 가능
- 근무사 : 한영 및 숫자 2~30자 입력 가능
- 역할 : 개발,운영,분석설계,PL 외 다수 항목에 한해서 선택 가능
- 기종 : ASUS, DELL, DELL EML 외 다수 항목에 한해서 선택 가능
- OS : LINUX, MAX, Ubuntu 외 다수 항목에 한해서 선택 가능
- 언어 : Angular, Android, ASP 외 다수 항목에 한해서 선택 가능
- DBMS : Access, Cassandra, DB2 외 다수 항목에 한해서 선택 가능
- Tool : .Net, Android Studio, Eclips 외 다수 항목에 한해서 선택 가능

* 프로젝트명 입력 시, 동일 행 입력항목 필수 입력

4 취소/저장 버튼

- 취소 : 선택 시, 정규직 정보 검색 페이지로 이동
- 저장 : 선택 시, 유효성 검사 실행 후

통과할 경우 : 정규직 정보 검색 페이지로 이동

통과못할 경우 : 미통과 항목에 초점 제공 및 경고문구 표시

정규직 정보 검색• 01

The screenshot shows a search interface with the following numbered components:

- 1: Top-left search bar.
- 2: 'or' dropdown menu.
- 3: '검색' (Search) button with an exclamation mark icon.
- 4: '초기화 후 전부검색' (Reset then full search) button.
- 5: '[정규직] 입력' (Input for Regular Job) button.
- 6: '근무자/퇴사자' (Employee/Retiree) checkbox.
- 7: '개발년차' (Development Year) input field with dropdown arrows for year selection.
- 8: '자격증' (Qualification) section with checkboxes for '정보처리기사', '정보처리산업기사', and '비움' (None).
- 9: '소유기술' (Owned Technologies) section with dropdown menus for 기종 (Type), OS, 언어 (Language), DBMS, and tool.
- 10: '역할' (Role) section with checkboxes for 개발 (Development), 운영 (Operation), 분석설계 (Analysis/Design), PL (Project Leader), PM (Project Manager), and 웹디 (Web Design).
- 11: '최종학력' (Final Education) section with checkboxes for 고졸 (High School), 전문대졸 (Associate's Degree), 일반대학사졸 (Bachelor's Degree), 일반대석사졸 (Master's Degree), 일반대박사졸 (Doctorate), and 비움 (None).
- 12: '졸업학부' (Graduation Major) section with checkboxes for various fields of study: 컴퓨터공학 (Computer Engineering), 공과대 (Engineering), 자연대 (Natural Sciences), 경상대 (Social Sciences), 사회대 (Humanities), 인문대 (Humanities), 법대 (Law), and 기타 (Others).

1 키워드 검색

- 프리랜서 검색과 동일

2 or and

- 프리랜서 검색과 동일

3 검색 버튼

- 프리랜서 검색과 동일

4 초기화 후 전부검색 버튼

- 프리랜서 검색과 동일

5 [정규직] 입력 버튼

- 선택 시, 선택 시, 정규직 정보 입력 페이지로 이동

6 근무자/퇴사자

- 다중선택 체크박스 가능

7 개발년차

- 최소 0 ~ 최대 20년차 선택 가능

8 자격증

- 프리랜서 검색과 동일

9 소유기술

- 기종, OS, 언어, DBMS, tool에 한해서 선택 가능

10 역할

- 프리랜서 검색과 동일

11 최종학력

- 프리랜서 검색과 동일

12 졸업학부

- 다중선택 체크박스 가능

13 비움 버튼

- 선택 시, 해당 행에 입력된 값 모두 초기화

정규직 정보 검색[•] 02

정규직 정보검색 01 페이지 참고

[정규직] 입력

초기화 후 전부검색 비움 검색

1 역할
 개발 운영 분석설계 PL PM 웹디

2 최종학력
 고졸 전문대졸 일반대학사졸 일반대학석사졸 일반대학박사졸 비움

3 졸업학부
 컴퓨터공학 공과대 자연대 경상대 사회대 인문대 법대 기타

4 생일
월 ~ 월

5 나이
살~ (년생 ~ 년생)

6 성별
 남 여

7 졸업년도
년 ~ 년

1 역할

- 프리랜서 검색과 동일

2 최종학력

- 프리랜서 검색과 동일

3 졸업학부

- 다중선택 체크박스 가능

4 생일

- 최소 1 부터 12 선택 가능

5 나이

- 프리랜서 검색과 동일

6 성별

- 프리랜서 검색과 동일

7 졸업년도

- 프리랜서 검색과 동일

정규직 정보 검색• 03

정규직 정보검색 01 페이지 참고

[정규직] 입력

비율 검색

초기화 후 전부검색

1 독해 외국어

2 회화 외국어

3 소주주량

5 맥주주량

6 병역

or

1 독해 외국어

- 다중선택 체크박스 가능

2 회화 외국어

- 다중선택 체크박스 가능

3 소주주량

- 최소 0 부터 3 선택 가능

4 맥주주량

- 최소 0 부터 3 선택 가능

5 병역

- 프리랜서 검색과 동일

- 1 독해 외국어
- 다중선택 체크박스 가능

- 2 회화 외국어
- 다중선택 체크박스 가능

- 3 소주주량
- 최소 0 부터 3 선택 가능

- 4 맥주주량
- 최소 0 부터 3 선택 가능

- 5 병역
- 프리랜서 검색과 동일

정규직 정보 검색[•] 04

1 검색개수 : 4개 2 10 행 보기

3

이름 오름차순 출생년도 오름차순 나이 오름차순 주민번호 오름차순 최종학력 오름차순 최종학력졸업일 오름차순 입사일 오름차순

4

근무여부	번호	이름	성별	출생년도	나이	핸드폰	이메일	주민번호	거주지	최종학력	최종학력 졸업일	입사일	소유스킬	자격증
근무자	4	김소연	여	1985	39	01045128865	ksy0831@nav...	850831-23598...	서울시 강남구 ...	전문대 졸업	2008년 02월	2023년 11월 0...	C#,C++,Git,Le...	정보처리기사
근무자	3	김태현	남	1962	62	01054264815	kth0807@gm...	620807-13575...	서울시 관악구 ...	석사 졸업	1987년 02월	2023년 11월 0...	ASUS,C,GDB,I...	정보처리기사,정...
근무자	2	신근수	남	1997	27	01024859942	geunsu@dau...	971010-13446...	경기도 안양시...	고졸	2016년 02월	2023년 11월 0...	Android,HTML...	정보처리기사,정...
근무자	1	안소민	여	1995	29	01086958937	242010@nave...	951007-23541...	인천시 부평구 ...	전문대 졸업	2018년 02월	2023년 11월 0...	DELL,Linux,Or...	없음

6

<< < 1 > >>

1 검색개수

- 검색 결과 개수 표시

2 N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

3 정렬기준

- 검색결과에 따른 정렬 기준 선택 제공

4 검색결과

- 번호,스킬분야,이름,성별,나이,학력, 가상개발년차, 파견가능일, 파견가능일 확정여부, 핸드폰, 이메일, 전화상담일, 이력서 확인일 정보를 포함한 검색결과 기본 10개 표시

5 검색결과 상세

- 검색 결과 행 선택 시, 상세 정보 팝업 표시

6 페이지 이동

- 최소,최대 페이지 바로가기 및 최대 페이지 표시. 번호 선택 시, 해당 페이지로 이동

정규직 상세 팝업

1 정규직 개발자 상세 정보

아이디: dutkoo12 | 암호: | 암호확인: | (-없이 입력)

이름: 김소연 | 주민번호: 850831-2359814 | 핸드폰: 01045128865

이메일: ksy0831@naver.com | 병역: 연체 (여성은 연체로 입력해 주세요.)

[주소]: 서울시 강남구 논현동

긴급연락처(사고 대비) [연락 받을 사람 이름]: 김병철

[관계]: 부모 (부모, 형제, 친척, 친인)

[전화번호]: 01024518845

급여통장번호: 국민은행 3333020415188 (-없이 입력)

학력

학교명	학부	전공기록	부전공기록	졸업일
서울과학고등학교	고등학교 졸	공과대	환경공학	2008-02-11
영지전문대학	전문대 졸(대학 졸)			2008-02-11
	대학교 학사 졸□ 학원			연도·월·일
	대학교 석사 졸			연도·월·일
	대학교 박사 졸			연도·월·일

자격증

정보처리기사 정보처리산업기사

외국어 (중급 이상만 입력)

독해 가능 외국어: 영어, 일어, 중국어, 불어, 독일어
 회화 가능 외국어: 영어, 일어, 중국어, 불어, 독일어

IT 교육센터

교육센터명	교육기간	교육수료일	수료기록
한국소프트웨어인재개발원	6 개월	2013-02-21	필기합격
기초			
하루 훈련량: 1 ▾ 일	소후 주평: 1 ▾ 영	학수 후방 (500CC 기준): 3 ▾ 전	관심분야: <input checked="" type="checkbox"/> 창작 <input checked="" type="checkbox"/> 경제 <input checked="" type="checkbox"/> 사회 <input checked="" type="checkbox"/> 문화
개별경력 (최근 1건만 입력)			
프로젝트명 / 기간	고객사 / 근무사 / 역할	개발환경: 모바일 비주기	
프로젝트명: 클라우드솔루션	고객사: 로데마스팅	기종: 음성 ▾	
시작일: 2014년 02월	근무사: 풍데마스팅	OS: 윈도우 ▾	
종료일: 2016년 02월	역할: 팀리더	언어: 음성 ▾	
작업일: 2019년 02월	파견단계: 파견종료	DBMS: 음성 ▾	
목차일: 2021년 05월		tool: 음성 ▾	

2 닫기 3 삭제 4 수정

1 정규직 개발자 상세 정보 표시

- 검색결과에서 선택된 행에 상세정보를 팝업으로 표시 및 수정 비활성화

2 닫기 버튼

선택 시, 팝업 닫힘

3 삭제 버튼

- 선택 시, '정말 삭제하시겠습니까?' 알림 표시 후 확인 선택 시, '프리랜서 정보가 삭제되었습니다.' 표시 및 해당 정보 삭제되며 팝업 닫힘

4 수정 버튼

- 선택 시, [완료] 버튼으로 문구 변경되며 이름 입력칸으로 초점 이동 및 모든 입력칸 수정 활성화

프로젝트 정보 입력

프로젝트 정보

1 접수일자 연도-월-일 오늘 날짜로

기간 연도-월-일 ~ 연도-월-일

프로젝트명 비움

고객사 비움

2 개발인력 년차 명 정보처리기사필수 정보처리산업기사필수

운영인력 년차 명 정보처리기사필수 정보처리산업기사필수

퍼블리셔인력 년차 명 정보처리기사필수 정보처리산업기사필수

디자이너인력 년차 명 정보처리기사필수 정보처리산업기사필수

3 업무내용 비움

업무장소 비움

출근시간 시 분

협력업체

기타 비움

① 프로젝트 기본 정보 입력

- 접수일자 선택

* [오늘 날짜로] : 선택 시, 현재 년-월-일-시간-분 표시기간 : 최소치 날짜도 검색 가능

- 기간 : 최소기간은 최대기간 초과 불가
- 프로젝트명 : 한글 또는 영어 2~15자 입력가능
- 고객사 : 한글 또는 영어 2~15자 입력가능

② 프로젝트 필요 인력 입력

- 필요 인력 : 최대 0~99명 선택 가능
- 년차 : 최대 0~20년 선택 가능
- 자격증 : 체크박스 다중 선택 가능

③ 프로젝트 업무 관련 정보 입력

- 업무 내용 : 2~50 자 입력 가능
- 업무 장소 : 2~50 자 입력 가능
- 출근 시간 : [시] = 0~24시 선택 가능 / [분] = 00, 30 선택 가능
- 협력업체 : 네이버, 카카오, 쿠팡, 라인, APPLE 등 선택 가능

! 비움

- 선택 시, 해당 행에 입력된 값 모두 초기화

프로젝트 정보 검색• 01

The screenshot shows a search interface with the following numbered components:

- 1: 초기화 후 전부검색 (Clear after search all)
- 2: OR dropdown menu
- 3: 비움 (Empty) button
- 4: 검색 (Search) button
- 5: [프로젝트] 입력 (Project Input) button
- 6: 파견 시작일 (Deployment Start Date) input field with date range and 비움 (Empty) button
- 7: 파견 종료일 (Deployment End Date) input field with date range and 비움 (Empty) button
- 8: 파견기간 (Deployment Period) input field with date range and 비움 (Empty) button
- 9: 개발년차 (Development Year) dropdown menus for start and end years, and 비움 (Empty) button
- 10: 운영년차 (Operation Year) dropdown menus for start and end years, and 비움 (Empty) button
- 11: 디자이너연차 (Designer Year) dropdown menus for start and end years, and 비움 (Empty) button
- 12: 퍼블리셔연차 (Publisher Year) dropdown menus for start and end years, and 비움 (Empty) button
- 13: 자격증 (Qualification) dropdown menus for start and end years, and 비움 (Empty) button
- 14: 첨부파일 (Attached File) checkboxes: 정보처리기사필수, 정보처리산업기사필수, 모두선택 (All Selection), and 비우기 (Reset)
- 15: 접수일자 (Submission Date) input field with date range and 비움 (Empty) button

① ~ ④

- 위 검색 기능들과 동일

⑤ 비움 버튼

- 선택 시 해당 행에 입력된 값 초기화

⑥ [프로젝트] 입력 버튼

- 선택 시, 선택 시, 프리랜서 정보 입력 페이지로 이동

⑦ 파견 시작일 / 종료일

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

⑧ 파견 기간

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시
* 파견기간 확정 체크박스 선택 시 확정된 결과 출력

⑨ 인력년차

- 필수 인력 : 최대 0~99명 선택 가능
- 년 : 최대 0~20년 선택 가능

⑩ 자격증

- 다중선택 체크박스 가능
* [모두선택] 선택 시 체크박스 모두 선택

⑪ 접수일자

- 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

프로젝트 정보 검색• 02

3

4

5

6

1

2

프로젝트명	접수일	시작일	종료일	기간 (개월)	역할
devgru	2023년 11월 01일	2023년 11월 14일	2023년 12월 01일	1개월	개발 : 01명, 운영 : 04명, 웨디 : 02명, 피플리서 : 05명
광고웹플랫폼	2023년 06월 07일	2023년 08월 07일	2023년 11월 07일	4개월	개발 : 10명, 운영 : 10명, 웨디 : 05명, 피플리서 : 05명
기업모범사례	2023년 11월 01일	2023년 11월 20일	2024년 05월 17일	6개월	개발 : 02명, 운영 : 02명, 웨디 : 01명, 피플리서 : 02명
쇼핑몰웹플랫폼개발	2023년 05월 10일	2023년 06월 08일	2023년 10월 08일	5개월	개발 : 05명, 운영 : 04명, 웨디 : 05명, 피플리서 : 06명
애플리케이션보드구축	2023년 11월 07일	2023년 05월 12일	2023년 06월 28일	2개월	개발 : 08명, 운영 : 02명, 웨디 : 0명, 피플리서 : 0명
우리농산물웹플랫폼	2023년 06월 07일	2023년 08월 01일	2023년 11월 25일	4개월	개발 : 10명, 운영 : 05명, 웨디 : 05명, 피플리서 : 05명
웹 플랫폼 유지보수	2022년 11월 05일	2022년 11월 27일	2023년 05월 07일	6개월	개발 : 10명, 운영 : 0명, 웨디 : 0명, 피플리서 : 0명
웹플랫폼개발	2023년 03월 08일	2023년 04월 09일	2023년 11월 16일	8개월	개발 : 10명, 운영 : 01명, 웨디 : 13명, 피플리서 : 07명
저기요	2023년 01월 07일	2023년 05월 09일	2023년 09월 08일	5개월	개발 : 05명, 운영 : 10명, 웨디 : 02명, 피플리서 : 10명
지도웹플랫폼개발	2023년 05월 01일	2023년 06월 07일	2023년 11월 16일	6개월	개발 : 10명, 운영 : 09명, 웨디 : 10명, 피플리서 : 09명

검색개수 : 12개 10 행 보기

1 검색개수
• 검색 결과 개수 표시

2 N 행보기
• 10, 15, 20 개수에 한해 행보기 선택 제공

3 정렬기준
• 검색결과에 따른 정렬 기준 선택 제공

4 검색결과
• 프로젝트명, 접수일, 시작일, 종료일, 기간(개월) 역할 정보를 포함한 검색결과 기본 10개 표시

5 검색결과 상세
• 검색 결과 행 선택 시, 상세 정보 팝업 표시

6 페이지 이동
• 최소,최대 페이지 바로가기 및 최대 페이지 표시. 번호 선택 시, 해당 페이지로 이동

프로젝트 상세 팝업

1

프로젝트 정보 입력

프로젝트 정보

접수일자 2023-06-20 오늘 날짜로

기간 2023-07-03 ~ 2023-12-29

프로젝트명 테크노저널리스트 비움

고객사 롯데홈쇼핑 비움

개발인력 07 년차 03 명 정보처리기사필수 정보처리산업기사필수

운영인력 04 년차 03 명 정보처리기사필수 정보처리산업기사필수

퍼블리셔인력 02 년차 01 명 정보처리기사필수 정보처리산업기사필수

디자이너인력 02 년차 01 명 정보처리기사필수 정보처리산업기사필수

업무내용 개발자정보관리 비움

업무장소 가산한라원엔원타워 비움

출근시간 09 시 30 분

협력업체 업체1

기타 기타 비움

2 닫기 3 삭제 4 수정]

1 프로젝트 상세 정보 표시

- 검색결과에서 선택된 행에 상세정보를 팝업으로 표시 및 수정 비활성화

2 닫기 버튼

선택 시, 팝업 닫힘

3 삭제 버튼

- 선택 시, '프로젝트 정보가 삭제되었습니다.' 알림 표시 및 해당 정보 삭제되며 팝업 닫힘
변경된 프로젝트 검색 결과물은 팝업 닫힘과 함께 갱신되어 프로젝트 정보 검색 화면에 표시됨

4 수정 버튼

- 선택 시, [완료] 버튼으로 문구 변경되며 모든 입력칸 수정 활성화

파견 정보 입력^{• 01}

프로젝트 정보 검색

1 **파견 시작일**
연도-월-일 ~ 연도-월-일 비움

2 **파견 종료일**
연도-월-일 ~ 연도-월-일 **파견종료일확정** 비움

3 **파견기간**
개월~ 개월 비움

4 **역할**
 개발 운영 퍼블리셔 웹디

5 **개발년차**
년차~ 년차

6 **운영년차**
년차~ 년차

① 파견시작일

- 선택 시, 시스템 캘린더 표시

② 파견종료일

- 선택 시, 시스템 캘린더 표시

③ 파견기간

- 최소, 최대 0~12개월 선택 가능

④ 역할

체크박스 선택사항

⑤ 개발년차

- 최소, 최대 0~20년차 선택 가능

⑥ 운영년차

- 최소, 최대 0~20년차 선택 가능

파견 정보 입력 02

프로젝트 정보 검색

1 디자이너연차

2 퍼블리셔연차

3 자격증

4 접수일자

5 키워드

6

7 검색

8 초기화 후 전부검색

연도-월-일 ~ 연도-월-일 비움

정보처리기사필수 정보처리산업기사필수 모두선택 비우기

① 디자이너연차

- 최소, 최대 0~20년차 선택 가능

② 퍼블리셔연차

- 최소, 최대 0~20년차 선택 가능

③ 자격증

체크박스 선택사항

④ 접수일자

- 선택 시, 시스템 캘린더 표시

⑤ 키워드

- 키워드 검색 입력칸 : 한영,숫자 30자에 한해 입력 가능

⑥ Or and

- 두가지 키워드 사이 or, and 구분하여 검색 가능하도록 옵션 제공

⑦ 검색 버튼

- 선택 시, 검색 결과 표시

⑧ 초기화 후 전부검색 버튼

- 사용자 입력 검색 조건 초기화 후 모든 검색 결과 표시

파견 정보 입력• 03

The screenshot shows a search results table with the following columns: 프로젝트명 (Project Name), 접수일 (Received Date), 시작일 (Start Date), 종료일 (End Date), 기간 (개월) (Duration (Months)), and 역할 (Role). The table contains 10 rows of data. At the top right, there are buttons for 검색개수 (Search Count) set to 14개, 행 보기 (View Rows) set to 10, and a dropdown menu. At the bottom center, there is a navigation bar with page numbers and arrows.

프로젝트명	접수일	시작일	종료일	기간 (개월)	역할
devgru	2023년 11월 01일	2023년 11월 14일	2023년 12월 01일	1개월	개발 : 01명, 운영 : 04명, 웹디 : 02명, 퍼블리서 : 05명
광고웹플랫폼	2023년 06월 07일	2023년 08월 07일	2023년 11월 07일	4개월	개발 : 10명, 운영 : 10명, 웹디 : 05명, 퍼블리서 : 05명
기업모법사례	2023년 11월 01일	2023년 11월 20일	2024년 05월 17일	6개월	개발 : 02명, 운영 : 02명, 웹디 : 01명, 퍼블리서 : 02명
쇼핑몰웹플랫폼개발	2023년 05월 10일	2023년 06월 08일	2023년 10월 08일	5개월	개발 : 05명, 운영 : 04명, 웹디 : 05명, 퍼블리서 : 06명
애플리케이션보드구축	2023년 11월 07일	2023년 05월 12일	2023년 06월 28일	2개월	개발 : 08명, 운영 : 02명, 웹디 : 0명, 퍼블리서 : 0명
온라인 게임 플랫폼	2023년 06월 21일	2023년 06월 06일	2023년 11월 01일	5개월	개발 : 03명, 운영 : 07명, 웹디 : 03명, 퍼블리서 : 03명
우리농산물웹플랫폼	2023년 06월 07일	2023년 08월 01일	2023년 11월 25일	4개월	개발 : 10명, 운영 : 05명, 웹디 : 05명, 퍼블리서 : 05명
웹 플랫폼 유지보수	2022년 11월 05일	2022년 11월 27일	2023년 05월 07일	6개월	개발 : 10명, 운영 : 0명, 웹디 : 0명, 퍼블리서 : 0명
웹플랫폼개발	2023년 03월 08일	2023년 04월 09일	2023년 11월 16일	8개월	개발 : 10명, 운영 : 01명, 웹디 : 13명, 퍼블리서 : 07명
유기동물관련웹플랫폼	2023년 02월 08일	2023년 04월 03일	2023년 07월 07일	4개월	개발 : 05명, 운영 : 05명, 웹디 : 05명, 퍼블리서 : 05명

① 검색 개수

- 검색 결과 개수 표시

② N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

③ 정렬기준

- 검색결과에 따른 정렬 기준 선택 제공

④ 검색결과

- 검색결과 기본 10개 표시

⑤ 페이지 이동

- 최소, 최대 페이지 바로가기 및 최대 페이지 표시.
번호 선택 시, 해당 페이지로 이동
- 선택 시, 팝업 닫힘

⑥ 닫기 버튼

파견 정보 입력 04

프로젝트 파견 등록

선택된 프로젝트 : devgru

재검색

현재 투입 중인 개발자 : 4 명

역할	개발자명	핸드폰	이메일	현재년차	등록/삭제
개발	안소민	01086958937	242010@naver.com	0년0개월	등록
개발	김소연	01045128865	ksy0831@naver.com	0년0개월	등록
publisher	신근수	01024859942	geunsu@daum.net	0년0개월	삭제
운영	최유리	01042516312	glasschoi@naver.com	0년0개월	등록
PM	김태현	01054264815	kth0807@gmail.com	0년0개월	등록
개발	채병장	01080812521	cbh0922@naver.com	06년06개월	등록
개발	주윤린	01026262678	tuxukm8450@naver.com	10년10개월	등록
운영	영연우	01072086266	jfflx9446@naver.com	10년10개월	등록
분석설계	마지연	01022556754	ddf007@daum.net	10년10개월	등록

닫기

① 선택된 프로젝트 ② 재검색 버튼

- 선택된 프로젝트명 표시
- 선택 시, 기존 선택된 프로젝트 선택 해제 및 파견등록 02 페이지로 이동

③ 현재 투입 중인 개발자

- 선택된 프로젝트에 투입된 개발자 명수 표시

④ 검색결과 상세

- 선택된 프로젝트에 투입된 및 투입 가능 개발자 목록 스크롤 표시

⑤ 등록/삭제 버튼

- 선택 시, 등록 상태에 따라 문구 변경(등록>삭제)

파견 현황 검색• 01

정규직 정보 검색 01 페이지 참고

비움 검색

초기화 후 전부검색 파견 등록

1 근무자/퇴사자
 근무자 퇴사자

2 파견기간
개월~ 개월

3 파견시작일
년 월 ~ 년 월 비움

4 파견종료일
년 월 ~ 년 월 종료일확정 종료일미정정 비움

5 년차
년차~ 년차

6 역할
 개발 운영 분석설계 PM PL 웹디 기타 비움

① 근무자/퇴사자
• 체크박스 선택사항

② 파견기간
• 최소, 최대 0~12개월 선택 가능

① 파견시작일
• 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

② 파견종료일
• 최소 년-월-일 및 최대 년-월-일 시스템 캘린더 표시

① 년차
• 최소,최대 0~20년차 선택 가능

② 역할
체크박스 선택사항

파견 현황 검색[•] 02

1 검색개수 : 7개 2 10 행 보기

3 파견개발자명 오름차순 ▼ 프로젝트명 오름차순 ▼ 현재년차 내림차순 ▼ 파견기간 내림차순 ▼ 파견기간 오름차순 ▼ 파견기간 오름차순 ▼ 파견기간 오름차순 ▼

4	파견 개발자명	핸드폰	이메일	프로젝트명	현재년차	역할	파견기간 (개월)	파견 시작일	파견 종료일	종료일 카운트	협력업체
	김정학	01046143386	wjdgkr3386@gmail.com	devgru	10년10개월	개발	1개월	2023-11-14	2023-12-01	23일	어찌구
	신근수	01024859942	geunsu@daum.net	devgru	0년0개월	publisher	1개월	2023-11-14	2023-12-01	23일	어찌구
	전진성	01062241684	esglbd6525@naver.com	헬스케어앱개발	10년10개월	기타	1개월	2023-11-01	2023-12-01	23일	저찌구
	주소윤	01083757445	hpyrcf1877@naver.com	devgru	10년10개월	PL	1개월	2023-11-14	2023-12-01	23일	어찌구
	주운린	01026262678	tuxukm8450@naver.com	애플리케이션보드구축	10년10개월	개발	2개월	2023-05-12	2023-06-28	-133일	저찌구
	채병장	01080812521	cbh0922@naver.com	애플리케이션보드구축	06년06개월	개발	2개월	2023-05-12	2023-06-28	-133일	저찌구
	하서연	01023423557	seoyeon77@gmail.com	devgru	10년10개월	운영	1개월	2023-11-14	2023-12-01	23일	어찌구

5 << < 1 > >>

① 검색개수

- 검색 결과 개수 표시

② N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

③ 정렬기준

- 검색결과에 따른 정렬 기준 선택 제공

④ 검색결과

- 파견개발자명, 핸드폰, 이메일, 프로젝트명, 현재년차, 역할,
- 파견기간/시작일/종료일, 종료일카운트, 협력업체 정보를 포함한 검색 결과 표시

⑤ 페이지 이동

- 최소, 최대 페이지 바로가기 및 최대 페이지 표시. 번호 선택 시, 해당 페이지로 이동

결재 정보 검색

결재

The screenshot shows a search interface for decision information. At the top, there is a search bar (1) with a placeholder '검색어 입력' (Search term input), a search button (2) labeled '검색' (Search), and a button (3) labeled '[결재 작성]' (Create decision). Below the search bar is a navigation bar with buttons: '전체 1 건' (4), '전체' (5), '미결' (Incomplete), '승인' (Approved), '반려' (Rejected), and a dropdown menu '10 행보기' (10 travel logs) (6). The main area displays a single search result (7) for a document titled '연차 보고의 건' (Annual leave report case) by '정선희 사장' (President Jeong Seon-hee) on '2023-11-09 오전 11:13' (November 9, 2023, 11:13 AM). The result includes tabs for '진행상황' (Status), '제목' (Title), '결재자' (Approver), and '등록일' (Registration date). At the bottom, there is a page navigation bar (8) with buttons for '«<' (Previous page), '1' (Current page), and '>>' (Next page).

① 검색어 입력

- 한영,숫자 30자에 한해 입력 가능

② 검색 버튼

- 선택 시, 키워드에 한해 검색 결과 표시

③ [결재 작성] 버튼

- 선택 시, 결재 작성 페이지로 이동

④ 검색 개수

- 검색 결과 개수 표시

⑤ 결재 카테고리

- 전체,미결,승인,반려에 한해 카테고리 선택 제공
- 각 카테고리 선택 시, 카테고리별 결재 목록 표시

⑥ N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

⑦ 검색결과 상세

- 검색 결과 행 선택 시, 결제 상세 페이지로 이동

⑧ 페이지 이동

- 최소,최대 페이지 바로가기 및 최대 페이지 표시.
- 번호 선택 시, 해당 페이지로 이동

결재 상세 화면^(미결)

결재 내용 상세보기

1 등록일 : 2023-11-09 16:49
결재상태 : 미결
요청자 : 정찬성
제목 : 근태 보고의 건

2 내용 : ■ 일자 : 2023-11-13

■ 인원 현황 :
전지범
이주성
김우길
안승민
박재영
최유리
안해찬
신비한

■ 근태 상세내역 :
이주성 - 건강악화로 인한 2023-11-09 결근

■ 특이사항 :
없음

3 목록

4 수정/삭제

1 결재 정보

- 등록일 : 결재 정보 등록일 표시
- 결재상태 : 결재 상태에 따라 표시(승인, 미결, 보류)
- 요청자 : 요청자 이름 표시
- 제목 : 제목 표시

2 내용

- 검색결과에서 선택된 행에 상세정보 표시

3 목록 버튼

- 선택 시, 결재 목록 페이지로 이동

4 수정/삭제 버튼

- 미결건에 한해 버튼 표시
- 선택 시, 결재 내용 수정/삭제 페이지로 이동

결재 상세 화면

결재 내용 상세보기

1 등록일 : 2023-11-09 17:05

결재상태 : 승인

결재일시 : 2023-11-10 14:34

요청자 : 서나람

제목 : 연차 보고의 건

2 ■ 성명 : 서나람

■ 직급 : 대리

■ 연락처 : 01034541034

■ 신청기간 : 2023-11-13 ~ 2023-11-15

■ 사유 : 지병으로 인한 수술 (위염)

사기 보이오 위와 같은 사유로 보고서를 제출하오니 허락하여 주시기를 바랍니다.

3 목록

1 결재 정보

- 등록일 : 결재 정보 등록일 표시
- 결재상태 : 결재 상태에 따라 표시(승인, 미결.보류)
- 결재일시 : 승인/반려한 시각 표시
- 요청자 : 요청자 이름 표시
- 제목 : 제목 표시

2 내용

- 검색결과에서 선택된 행에 상세정보 표시

3 목록 버튼

- 선택 시, 결재 목록 페이지로 이동

결재 정보 입력

결재 입력

결재 정보

1 문서 종류 : 지출보고서

2 결재자 : 정선희 사장

제목 : 지출 보고의 건

내용

■ 작성 일시 :

■ 금 액 :

■ 지출 내용 :

상기와 같이 지출 내용을 보고합니다.

20 년 월 일

3 취소 4 저장

① 결재 정보

- 문서 종류 : 자유양식 기본 선택으로 지출보고서, 업무보고서, 영업보고서, 연차보고서, 근태보고서에 한해 선택 제공 및 각 문서 종류별 상이한 양식 표시
- 결재자 : 최종 결재자 이름 표시
- 제목 : 최대 100자 입력 가능

② 내용

- 최대 2000자 입력 가능

③ 취소 버튼

- 선택 시, 결재 목록 페이지로 이동

④ 저장 버튼

- 선택 시, 작성내용 저장되며 결재 목록 페이지로 이동

공지사항 검색

공지사항

1 2 3

4 친체 21 건 5 10 행보기

6

번호	제목	작성자	날짜	조회수
21	신입 프리랜서 입사	권용순	2023-11-07	7
20	신입 프리랜서 입사	권용순	2023-11-07	8
19	신입 프리랜서 입사	권용순	2023-11-07	3
18	신입 프리랜서 입사	권용순	2023-11-07	3
17	신입 프리랜서 입사	권용순	2023-11-07	1
16	신입 프리랜서 입사	권용순	2023-11-07	1
15	신입 프리랜서 입사	권용순	2023-11-07	2
14	신입 프리랜서 입사	권용순	2023-11-07	2
13	신입 프리랜서 입사	권용순	2023-11-07	1
12	신입 프리랜서 입사	권용순	2023-11-07	2

7 << < 1 2 3 > >>

① 검색어 입력

- 한영,숫자 30자에 한해 입력 가능

② 검색 버튼

- 선택 시, 키워드에 한해 검색 결과 표시

③ 새글쓰기 버튼

- 선택 시, 공지사항 작성 페이지로 이동.
- 관리자 계정에 한해 표시

④ 검색개수

- 검색 결과 개수 표시

⑤ N 행보기

- 10, 15, 20 개수에 한해 행보기 선택 제공

⑥ 검색결과 상세

- 검색 결과 행 선택 시, 공지사항 상세 페이지로 이동

⑦ 페이지 이동

- 최소,최대 페이지 바로가기 및 최대 페이지 표시.
번호 선택 시, 해당 페이지로 이동

공지사항 입력

공지사항 작성

제목 ①

작성자 ②

본문 내용은 최대 500자 입력
③

[파일 선택] 선택된 파일 없음 ④

저장 ⑤

목록 ⑥

This screenshot displays a user interface for creating a notice. At the top, there are input fields for '제목' (Title) and '작성자' (Author). Below these is a large text area for the '본문' (Body) with a character limit of 500. At the bottom, there are buttons for file selection, saving, and viewing the list of notices.

① 제목

- 최대 30자 입력 가능

② 작성자

- 최대 30자 입력 가능

③ 본문

- 최대 500자 입력 가능

④ 파일선택 버튼

- 선택 시, 저장장치의 단일 파일(이미지, 엑셀) 선택 가능

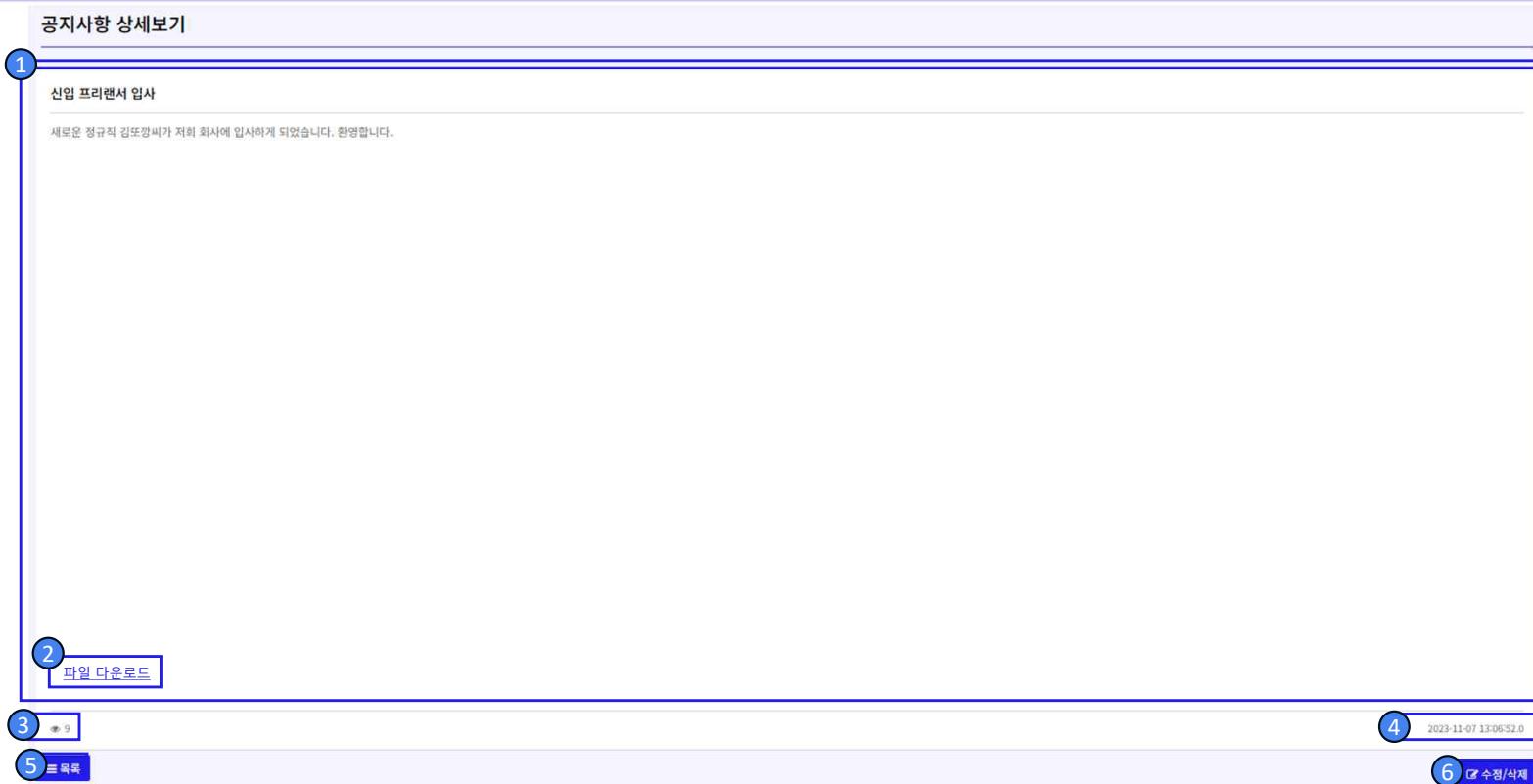
⑤ 저장 버튼

- 선택 시, 공지사항 수정/삭제 페이지로 이동

⑥ 목록 버튼

- 선택 시, 공지사항 목록 페이지로 이동

공지사항 상세 화면



① 공지사항 상세

- 사용자 입력 제목 및 본문 내용 표시

② 파일 다운로드 버튼

- 업로드 된 파일이 있을 경우 표시
- 선택 시, 로컬에 파일 다운로드 됨

② 조회수

- 공지사항 상세보기 조회수 표시

③ 작성일

- 공지사항 작성 상세 일자 표시(년월일시간분초)

④ 목록 버튼

- 선택 시, 공지사항 목록 페이지로 이동

⑤ 수정/삭제 버튼

- 선택 시, 공지사항 수정/삭제 페이지로 이동

05

프로젝트 시연 ·

06

문제 해결 ·

문제해결[•] 트랜잭션

문제

```
@Service  
@Transactional  
public class FreelancerServiceImpl implements FreelancerService{  
    @Autowired  
    private FreelancerDAO freelancerDAO;  
  
    @Override  
    public int insertFreeInfo( FreelancerDTO freelancerDTO ) throws Exception {  
        int freelancerRegCnt = 0;  
  
        freelancerRegCnt = this.freelancerDAO.insertFreeInfo(freelancerDTO);  
        freelancerRegCnt = this.freelancerDAO.free_can_work_location_info(freelancerDTO);  
        freelancerRegCnt = this.freelancerDAO.free_favorite_part_code(freelancerDTO);  
        if(freelancerDTO.getLicense_code() != null && freelancerDTO.getLicense_code().size() > 0) {  
            freelancerRegCnt = this.freelancerDAO.free_license(freelancerDTO);  
        }  
        freelancerRegCnt = this.freelancerDAO.free_skill_category(freelancerDTO);  
        freelancerRegCnt = this.freelancerDAO.free_year_info(freelancerDTO);  
        if(freelancerDTO.getBlack_reason_name() != null && freelancerDTO.getBlack_reason_name().size() > 0) {  
            this.freelancerDAO.is_black_info(freelancerDTO);  
        }  
  
        return freelancerRegCnt;  
    }  
}
```

Controller에서 호출한 serviceimpl 파일의 메소드들을 하나의 트랜잭션으로 오해하였으나, 실제로는 각 serviceimpl 메소드가 별개의 트랜잭션으로 처리되어 있었습니다. 이로 인해 WAS와의 통신 중 문제 발생 시 DB에는 데이터가 중간까지만 반영되고 남아 있었습니다.

해결

```
@Override  
public int insertFreeInfo( FreelancerDTO freelancerDTO ) throws Exception {  
    int freelancerRegCnt = 0;  
    freelancerRegCnt = this.freelancerDAO.insertFreeInfo(freelancerDTO);  
    return freelancerRegCnt;  
}  
  
@Override  
public int free_can_work_location_info(FreelancerDTO freelancerDTO) throws Exception{  
    int free_can_work_location_infoCnt = 0;  
    free_can_work_location_infoCnt = this.freelancerDAO.free_can_work_location_info(freelancerDTO);  
    return free_can_work_location_infoCnt;  
}  
@Override  
public int free_favorite_part_code(FreelancerDTO freelancerDTO) throws Exception{  
    int free_favorite_part_codeCnt = 0;  
    free_favorite_part_codeCnt = this.freelancerDAO.free_favorite_part_code(freelancerDTO);  
    return free_favorite_part_codeCnt;  
}  
@Override  
public int free_license(FreelancerDTO freelancerDTO) throws Exception{  
    int free_licenseCnt = 0;  
    if(freelancerDTO.getLicense_code() == null) {  
        free_licenseCnt++;  
        return free_licenseCnt;  
    } else {  
        free_licenseCnt = this.freelancerDAO.free_license(freelancerDTO);  
        return free_licenseCnt;  
    }  
}
```

이전에 각 서비스 메소드에 개별적으로 걸린 트랜잭션을 **하나의 트랜잭션으로 통합**하여 데이터베이스에 일괄 적용하도록 수정했습니다. 이를 통해 각 서비스 메소드의 실행이 모두 성공 혹은 전체가 실패하도록 수정했습니다. 이는 DAO 호출을 서비스 메소드 내부에 포함시켜 모든 데이터 작업을 동일한 트랜잭션으로 묶어 처리함으로써, 중간 단계에서의 문제 발생을 방지하고 **데이터 일괄성을 유지하도록 했습니다.**

문제해결• 알리아스 불일치

문제

```
<select id="getDevList" parameterType="com.naver.erp.DevSearchDTO" resultType="java.util.HashMap">
    select * from( select rownum RNUM, zxvcv.* from(
        select
            d.dev_num "dev_num"
            ,dname
            ,(decode(substr(jumin_num, 7, 1), '1', '남', '3', '남', '여')) "gender"
            ,(case
                substr(jumin_num, 7, 1)
                when '1' then '19'
                when '2' then '19'
                else '20'
            end||substr(jumin_num, 1, 2)) "birth_year"
            ,(to_number(to_char(sysdate, 'YYYY')))
            -
            to_number(
            case
            substr(jumin_num,7,1)
            when '1' then '19'
            when '2' then '19'
            else '20'
            )
    )
```

근무여부	번호	이름	성별	출생년도	나이	핸드폰	이메일	주민번호	거주지	최종학력	최종학력 발급일	입사일	소유수입	자격증
근무자	33	[REDACTED]	남	1983	41	0104832112	crazy339@gmail...	830712-1115486	서울특별시 혁선...	고졸	2023년 10월	2023년 11월 09일	.Net,ASP,ASUS...	없음
근무자	32	[REDACTED]	남	1991	33	01087785232	woong@naver...	911015-1234698	서울특별시 영등...	학사 졸업	2017년 02월	2023년 11월 09일	.Net,ASP,Acc...	정보처리기사
근무자	31	[REDACTED]	여	1975	49	01042269584	mikyeyong@nav...	751122-2157541	인천 남동구 만화...	전문대 졸업	1998년 02월	2023년 11월 09일	Java,JavaScript	정보처리기사

데이터베이스 쿼리 클라이언트를 사용하여 쿼리를 직접 실행하였을때는 오류가 발생하지 않았지만 쿼리 결과를 프로그래밍 언어로 가져올 때 결과값을 가져오지 못하는 문제점이 발생했습니다.

해결

```
<select id="getDevList" parameterType="com.naver.erp.DevSearchDTO" resultType="java.util.HashMap">
    select * from( select rownum RNUM, zxvcv.* from(
        select
            d.dev_num "dev_num"
            ,dname
            ,(decode(substr(jumin_num, 7, 1), '1', '남', '3', '남', '여')) "gender"
            ,(case
                substr(jumin_num, 7, 1)
                when '1' then '19'
                when '2' then '19'
                else '20'
            end||substr(jumin_num, 1, 2)) "birth_year"
            ,(to_number(to_char(sysdate, 'YYYY')))
            -
            to_number(
            case
            substr(jumin_num,7,1)
            when '1' then '19'
            when '2' then '19'
            else '20'
            )
    )
```

근무여부	번호	이름	성별	출생년도	나이	핸드폰	이메일	주민번호	거주지	최종학력	최종학력 발급일	입사일	소유수입	자격증
근무자	33	박정희	남	1983	41	0104832112	crazy339@gmail...	830712-1115486	서울특별시 혁선...	고졸	2023년 10월	2023년 11월 09일	.Net,ASP,ASUS...	없음
근무자	32	이은기	남	1991	33	01087785232	woong@naver...	911015-1234698	서울특별시 영등...	학사 졸업	2017년 02월	2023년 11월 09일	.Net,ASP,Acc...	정보처리기사
근무자	31	박미경	여	1975	49	01042269584	mikyeyong@nav...	751122-2157541	인천 남동구 만화...	전문대 졸업	1998년 02월	2023년 11월 09일	Java,JavaScript	정보처리기사

쿼리에서 사용한 alias 나 컬럼이름이 프로그래밍 언어에서 사용한 이름과 일치하지 않는 경우, 결과를 가져오는데 문제가 발생할 수 있다는 것을 알아냈습니다. 쿼리에서 사용한 alias나 컬럼 이름과 프로그래밍 언어에서 사용하는 변수 또는 필드 이름을 일치시킴으로써 문제를 해결하였습니다.

문제해결• 다중행 처리

문제

```
updateDevCnt = this.devDAO.updateDevjuniorcollegeInfo(devDTO);
updateDevCnt = this.devDAO.updateDevUniversityInfo(devDTO);
updateDevCnt = this.devDAO.updateDevMasterInfo(devDTO);
updateDevCnt = this.devDAO.updateDevDoctorInfo(devDTO);
updateDevCnt = this.devDAO.updateDevItcenterInfo(devDTO);
updateDevCnt = this.devDAO.updateDevLikingInfo(devDTO);
updateDevCnt = this.devDAO.updateDevPastCompanyInfo(devDTO);
```

다중행으로 저장된 데이터를 수정하는 기능을 구현하는 과정중에 sql 구문 중 update를 사용하여 구현했으나 수정된 다중행의 신규값은 처리하지 못하고 DB에 잘못된 데이터가 잔여하는 에러가 발생하였습니다.

해결

: 수정 후 delete 구문

```
updateDevCnt = this.devDAO.delete_itCenter_info(devDTO);
updateDevCnt = this.devDAO.delete_dev_past_prj_info(devDTO);
updateDevCnt = this.devDAO.delete_doctor_info(devDTO);
updateDevCnt = this.devDAO.delete_juniorCollege_info(devDTO);
updateDevCnt = this.devDAO.delete_master_info(devDTO);
updateDevCnt = this.devDAO.delete_university_info(devDTO);
updateDevCnt = this.devDAO.delete_past_company_info(devDTO);
```

: 수정 후 insert 구문

```
if(devDTO.getJuniorCollege_name().length()!=0) {
    updateDevCnt = this.devDAO.juniorCollege_info(devDTO);
}
if(devDTO.getUniversity_name().length()!=0) {
    updateDevCnt = this.devDAO.university_info(devDTO);
}
if(devDTO.getMaster_name().length()!=0) {
    updateDevCnt = this.devDAO.master_info(devDTO);
}
if(devDTO.getDoctor_name().length()!=0) {
    updateDevCnt = this.devDAO.doctor_info(devDTO);
}
if(devDTO.getItCenter_name().length()!=0) {
    updateDevCnt = this.devDAO.itCenter_info(devDTO);
}
```

다중행으로 저장되는 데이터를 수정하기 위해서는 기존 데이터를 **update하는 방식이 아닌** delete를 통해 **초기화 후** insert를 통해 **재할당 해주는 방식으로** 변경해 해당 문제를 해결하였습니다.

문제해결• 타입 오류

문제

```
329•     public int getFree_uid() {  
330         return free_uid;  
331     }  
332  
333•     public void setFree_uid(int free_uid) {  
334         System.out.println(free_uid+"-> free_uid");  
335         this.free_uid = free_uid;  
336     }  
337  
338 }
```

Console X
prj - Pj[Application [Spring Boot App] C:\wts-4.17.0.RELEASE\plugins\org.eclipse.jdt.core\openjdk\hotspot\jre\full\win32\x86_64_17.0.5\20221102-0933\wts\blin\javaw.exe (2023-11-17 오전 10:44:07) [pid: 6832]
[Failed to convert property value of type 'java.lang.String' to required type 'int' for property 'free_uid'; nested exception is java.lang.NumberFormatException: For input string: "F10-> free_uid"]

변수의 타입과 다른 타입의 데이터가 대입되어 DB에 데이터를 요청 및 할당 할때 타입 불일치 오류로 화면에 데이터를 표시하시 못하거나 입력값을 DB로 보내지 못하는 경우가 종종 발생했습니다.

해결

```
328  
329•     public String getFree_uid() {  
330         return free_uid;  
331     }  
332  
333•     public void setFree_uid(String free_uid) {  
334         System.out.println(free_uid+"-> free_uid");  
335         this.free_uid = free_uid;  
336     }  
337  
338 }
```

Console X
prj - Pj[Application [Spring Boot App] C:\wts-4.17.0.RELEASE\plugins\org.eclipse.jdt.core\openjdk\hotspot\jre\full\win32\x86_64_17.0.5\20221102-0933\wts\blin\javaw.exe (2023-11-17 오전 10:44:07) [pid: 6832]
F10-> free_uid

각 데이터 타입별 요청 및 **할당할 타입**을 정확히 **파악해** 타입 불일치 **에러를 최소화했습니다.**

문제해결[•] 변수 불일치

문제

```
private String family_relation1;
private String family_name1;
private String family_job1;
private String family_birth_year1;
private String is_live_together1;
```

정규직 페이지의 데이터를 전송하려했지만 전송이 안되는 오류가 발생했습니다. 데이터베이스 쿼리 클라이언트를 사용하여 쿼리를 직접 실행하였을때는 오류가 발생하지 않아 alias 선언의 문제인줄 알았지만 그것 또한 아니었습니다.

해결

```
<div class="fam_desc desc_f">
<div>
<select name="family_relation1" id="family_relation">
<option value=""></option>
<option value="1">부</option>
<option value="2">모</option>
<option value="3">자녀</option>
<option value="4">형제자매</option>
<option value="5">배우자</option>
</select>
</div>
<div>
<input type="text" name="family_name1" maxlength="30">
</div>
<div>
<input type="text" name="family_job1" maxlength="30">
</div>
<div>
<select name="family_birth_year1">
<option value="n"></option>
<c:forEach var="i" begin="1910" end="2023">
<option value="${i}">${i}</option>
</c:forEach>
</select>년
</div>
<div>
<td><input type="checkbox" name="is_live_together1" value="동거">동거</td>
</div>
..
```

데이터가 전송되지 않는 경우중 하나가 변수명이 다를 경우 전송되지 안된다는 것을 인지했습니다. 해결 사항으로 DTO 클래스와 JSP 페이지의 input 요소의 **name을 동일하게 설정하고** DTO클래스에서 해당 name과 매핑되는 어노테이션을 사용하여 **문제를 해결하였습니다.**

문제해결[•] **undefined**

문제

```
else if( checkObj_dispatch_role.val() == null || checkObj_dispatch_role.val() == '' ) {  
    alert("개발경력 > 역할을 선택해 주세요.");  
    checkObj_dispatch_role.focus();  
    return;  
}
```

해결

```
else if( checkObj_dispatch_role.val() == undefined || checkObj_dispatch_role.val() == '' ) {  
    alert("개발경력 > 역할을 선택해 주세요.");  
    checkObj_dispatch_role.focus();  
    return;  
}
```

값이 NULL 일 경우를 미리 상정하고 쿼리를 구성하였지만, 결과값을 NULL로 인식을 못하는 것인지 아니면 NULL이 아닌 다른 값이 들어오는 것인지 들어오는 값을 처리를 못하는 문제점이 발생했습니다.

JavaScript에서 데이터베이스로부터 결과를 가져올 때, 'NULL' 대신 'undefined' 값을 받는 경우가 있다는 것을 확인했습니다.
조건문을 사용하여 값이 'undefined'인 경우에 대한 대체값을 설정해 문제를 해결하였습니다.

06

후기.

후기 문제해결

프로젝트를 시작하고 팀장을 맡은 순간, 처음에는 불안과 걱정이 앞섰습니다.

비전공자면서, 팀원 중에서 가장 어린 내가 팀을 끌어 나갈 수 있을지에 대한 불안감이 컸습니다.

그러나 팀원들의 호응과 협력으로 걱정했던 모든 것들이 성과로 이어졌습니다.

팀 전체가 자주 의견을 나누고 회의하면서 효율적인 의사소통과 열린 토론을 통해 목표를 공유하고 문제를 해결했습니다.

팀원들의 단합이 프로젝트의 성패를 좌우했다고 느꼈습니다.



팀장 정선희

“ 팀원간의 화합이 성패다! ”

프로젝트 도중, 웹 개발에 아직 익숙하지 않은 팀원들이 있었습니다. 팀원들에게 참고가 될 예시를 넘겨주기 위해 **프리랜서 파트의 입력/수정 JSP, Java, XML**들을 훈자 구현했습니다. 훈자 구현하는 과정에서 새로운 기술들을 습득하고 수업에서 배웠던 추상적인 개념들이 실무에서 어떻게 구체화하는지를 직접 체험할 수 있었습니다.

이후 역할 분담 방식을 변경했습니다. 각 팀원이 자신의 파트를 독립적으로 구현하도록 함으로써 목적은 배움에 더 중점을 두게 되었습니다.

부담스럽고 갑작스러운 변경이었지만 팀원들이 긍정적으로 받아들여 줘서 모두 함께 성장할 수 있었습니다.

하지만 예상치 못한 어려움도 있었습니다. 일정 지연 문제가 검색 조건이 복잡한 정규직 부분에서 발생했습니다. 정규직 담당 팀원에게 3일 안에 XML 파일에 들어갈 select 구문을 구상하도록 요청했지만, 생각보다 시간이 소요되었습니다. 3일 동안은 낙제한 시간이라고 생각했기에 일정이 밀린 것에 당황했습니다.

그러나 팀원을 질책하는 대신 해당 팀원을 도와 어려워했던 **select 구문 안 내용을 함께 구상**해 주면서 문제를 해결했습니다. 팀장이지만 프로젝트에 관련된 자세한 것들을 모두 알 수는 없다는 것과, 팀원들과의 소통을 통해 현실적인 일정을 공유하고 서로의 어려움을 이해하는 능력이 중요하다는 것을 깨달았습니다.

후기 문제해결

프로젝트를 완성하고 나서는 큰 성취감을 느꼈습니다.

수업에서 배운 내용을 실제 프로젝트에 적용하고, 미처 다루지 못한 새로운 기술들을 습득하는 과정에서 개발자로서의 역량을 향상시킬 수 있었습니다. 단순히 기술적인 부분뿐만 아니라 팀원들과의 협력을 통해 리더십, 의사소통, 문제 해결 능력 등 여러 면에서 성장할 수 있었습니다. 다만, 프로젝트를 마치면서 통계 페이지를 추가하지 못한 것이 아쉬움으로 남았습니다.

이는 사용자 편의성을 고려하지 않고 메인 기능에만 초점을 맞춤으로써 발생한 문제였습니다.

시간이 나면 통계 페이지를 추가하고, 향후 프로젝트에서는 사용자 편의 측면도 고려하여 프로젝트를 진행할 계획입니다.



“팀원간의 화합이 성패다!”

한 분야에만 집중하는 것이 아니라, 팀장으로서 팀원을 도와 여러 파트를 종합적으로 다루면서 추상적인 개념을 실제로 어떻게 적용하는지를 배울 수 있었습니다.

전공자가 아니라는 점에서 불안감을 느꼈지만, 팀원들과의 협업을 통해 서로의 강점을 살려가며 프로젝트를 성공적으로 진행할 수 있었습니다.

프로젝트를 진행하면서 **지속적인 학습의 중요성**을 깨닫게 되었습니다. 또한 어떤 것이 부족했는지를 알게 되면서 앞으로 어떤 방향으로 개발을 공부해야 할지에 대한 인사이트를 얻게 되었습니다.

프로젝트를 진행하고 성공적으로 마무리하면서 얻은 경험은 나에게 개발에 대한 큰 자신감과 동시에, 계속해서 성장해 나가야 한다는 동기부여를 주었습니다.

프로젝트에서 팀장 역할을 맡은 것이 큰 행운이라고 생각합니다. 팀장이었기 때문에 단순 개발의 실력을 늘린 것뿐만 아니라 책임감을 가지고 팀을 어떻게 끌어 나가야 하는지 많이 깨달았고, 그만큼 성장했습니다.

앞으로는 이러한 경험을 토대로 개발자로서의 경력을 쌓아나가며, 더 큰 규모와 복잡도를 가진 프로젝트를 성공적으로 완료시키는데 도전하고자 합니다.

또한, **지속적인 학습과 역량 강화를 통해 빠르게 변화하는 기술 트렌드에 발맞춰 선도적인 역할을 수행하는 개발자로 성장하고자 합니다.**

후기* 문제해결



“ 넥엔드로 새로운 도전! ”

후기 문제해결

프로젝트 초기에는 모든 파트가 이해하기 어려웠습니다. 특히, 데이터 흐름을 파악하는 데에 있어서 많은 난해한 부분이 있었으나, 팀원들과의 협업을 통해 극복해 나갈 수 있었습니다. 서로가 비슷한 어려움을 겪고 있다는 것을 알게 되면서, 함께 자료를 찾아보고 논의하며 이해를 돋기 시작했습니다. 팀 내에서의 커뮤니케이션은 매우 중요했습니다. 꾸준한 회의를 통해 함께 모여 구문을 이해하고 적용하는 방법을 토의하고 설명하는 시간을 가졌고, 서로의 관점을 이해하고 혼란스러운 부분을 함께 해결하면서 점차적으로 어려움을 극복해 나갔습니다.

특히, 누군가가 이해한 내용을 다른 팀원들에게 설명하면서, 더 깊이 있는 이해를 얻을 수 있었습니다.



팀원 안소민

“별도함이 그대로,”

가장 많은 시간을 할애했던 정규직 mapper 구문은 SI ERP 프로젝트에서 주요 도전 중 하나였습니다. 초기에는 이해하기 어려웠고, 구문의 복잡성으로 인해 불안감을 느꼈습니다. 하지만 이를 극복하기 위해 몇 가지 전략을 활용했습니다. 먼저, 친절하고 전문적인 도움을 받을 수 있는 커뮤니티나 온라인 자료를 찾아보았습니다.

관련된 포럼이나 커뮤니티에 참여하여 다른 사람들의 경험과 조언을 듣고, 자료를 찾아보며 개념을 이해하는 데 도움을 받았습니다. 또한, 프로젝트 팀 내에서 서로의 경험을 공유하고 협력하여 어려움을 극복하는 것이 중요했습니다. 특히, 자신이 이해한 부분을 다른 팀원에게 설명하면서 개념을 더욱 명확하게 이해할 수 있었습니다.

또한, 구문의 작동 방식을 실제 데이터와 연결하여 실험하고 테스트해보는 것이 중요했습니다. 예제를 활용하여 실제 데이터에 적용해보면서 오류를 발견하고 수정하는 과정을 거치면서 점차적으로 익숙해지고 자신감을 키울 수 있었습니다. 마지막으로, 시간과 노력을 기울여 연습하는 것이 필요했습니다. 어려운 구문을 이해하고 적용하는 데에는 시간이 걸리지만, 꾸준한 노력과 반복된 학습을 통해 점차적으로 능숙해질 수 있었습니다. 이러한 노력과 전략을 통해 정규직 mapper 구문의 어려움을 극복하고 프로젝트에 기여할 수 있었습니다.

이러한 과정을 통해 얻은 경험은 미래의 프로젝트에서도 유용하게 활용될 것이라 믿습니다. 팀원들과의 협업은 정말 소중한 경험이었습니다. 서로가 어려움을 겪으면서도 함께 문제를 해결하고자 하는 우리의 모습을 보면서 많은 것을 배웠습니다. 특히, 다양한 배경과 지식을 가진 팀원들과 함께 작업하면서 서로의 관점을 이해하고 존중하는 방법을 배웠습니다. 각자가 가진 강점을 살려 서로에게 도움을 주고 받으며, 그것이 프로젝트 성공으로 이어지는 것을 경험했습니다. 또한, 이러한 팀워크는 대인관계와 리더십 능력을 향상시키는 데에도 도움이 되었습니다. 어려움을 함께 극복하고 함께 성취하는 과정에서 팀의 결속력이 강화되었고, 서로를 신뢰하고 의지하는 것이 프로젝트 진행에 큰 영향을 미쳤으며, 이렇게 큰 성장을 이루도록 소중한 경험을 할 수 있게 해 준 팀원들에게 감사합니다.

후기 문제해결

프로젝트에 시작하기 앞서서 조원들 모두 첫 프로젝트였기 때문에 주제를 정하는 일도 굉장히 힘들었습니다.

SI 인력 관리 프로젝트를 선택한 이유는 여러 가지입니다. 우선, 이 프로젝트는 우리 조원들에게 개발자로서의 첫발을 내딛게 될 중요한 출발점이었습니다. 또한, 회사와 개발자, 그리고 현장 매니저 간의 역할과 관계에 대한 이해 뿐 아니라, 만약 이 프로젝트를 사용하는 클라이언트나 매니저라면 무엇을 원하고 어떻게 하는 것이 더 좋을지에 대한 고려가 가능한 주제라고 생각하여 선택하게 되었습니다.



“최적의 업무분담!”

이번 프로젝트에서 나의 역할은 DB에서 데이터를 받아 JAVA DTO 객체에 저장시키고 쿼리문에 데이터를 연동하여 결과값을 내는 역할을 맡았습니다.

교육만 듣다가 막상 무언가를 내가 스스로 만들려고 하니 굉장히 막막했지만 팀원들에게 물어가면서 하나하나 Java 코드를 작성하기 시작했습니다.

이번 프로젝트의 장점은 적절한 업무 분담이었다고 생각합니다. 조원 누구 하나 빠짐없이 프로젝트의 한 부분을 각각 맡아 했으며, 프로젝트 테스트 또한 누구 한 명이라도 빠짐없이 참여하여 협업에 대한 자신감이 생겼습니다. 6회 간의 회의를 통하여서 해야 할 일, 끝내지 못한 일, 개선해야 할 일 등 서로서로 피드백을 해주며 팀원 서로 간의 소통이 매우 원활했습니다. 또한 리더쉽을 경비한 팀장과 부팀장 덕분에 소통에 차질이 없었다고 생각됩니다.

단점으로는 주석의 부재가 있었습니다. 코드를 작성하면서 매번 주석을 적었어야 했지만 주석까지 신경을 쓸 여유가 없었습니다. 주석을 신경을 쓰지 않은 탓에 팀원의 코드를 보고 이해하기 힘들었던 점이 있었습니다. 또한 내가 작성했음에도 다시 한번 코드를 찬찬히 읽어보아야 이해가 되었고 도리어 코드를 수정하거나 추가할 때 시간이 더 많이 들었던 것 같습니다. 코드를 작성하면서 힘들지만 주석을 작성하는 습관을 들여야 나중에 고생을 안 한다는 점을 알 수 있었습니다.

프로젝트를 진행하면서 개발자로서 해야 할 역할을 실감하게 되었습니다. 코드 작성과 디버깅, 팀원들과의 프로젝트 협업은 귀중한 경험이었고, 이를 통해 개발자로서 첫발을 내디디며 조금이나마 성장할 수 있었습니다. 앞으로 더 많은 프로젝트를 경험하며 지식과 기술을 향상하고자 합니다.

후기 문제해결

처음 해보는 공동프로젝트에 두려움을 느꼈습니다. 역할을 정할 때 저는 자바 쪽을 맡기로 했습니다.

Controller로 받아서 DTO에 저장시키고 mapper까지 넘기는 작업을 했습니다.

제가 한 작업이 잘 작동하는지 확인할 때마다 생각지도 못한 곳에서 오류코드가 떴고 볼 때마다 막막했습니다. 계속해서 검색을 하면서 안되는 이유를 찾았습니다. 인상 깊었던 오류 중에 데이터 저장을 누르면 ORA-00936: missing expression , SQL 구문이 잘못되었다는

오류가 발생했습니다. mapper에서 한참을 찾다가 코드를 처음부터 따라가 봐야겠다고 생각해서 쭉 봤는데 jsp에 있는

입력양식인 에서 value를 넣지 않아서 생겼던 오류였습니다. 이런 어이없는 실수로 오래 걸릴 수도 있다고 느꼈습니다.

오류를 찾기 위해 JSP 와 mapper 까지 봐야 했고 어느 순간 jsp와 mapper까지 만지고 있는 저를 보았습니다.



“ 생각보다 할만한데? ”

주기적인 회의를 통해서 현재 상황과 다음 단계를 공유하며 문제를 해결하고 해야 할 일을 받았습니다. 차근차근 주어진 일을 하기 위해 jsp, DTO, DAO, mapper 등등 하나씩 작업하다 보니 코드의 흐름을 알게 되었고 어렵기만 했던 코딩이 조금씩 ‘할 만하네?’ 생각으로 바뀌게 되었습니다. 팀원들과 소통하면서 여러 가지 팁도 배웠습니다. 거의 2개월 동안 팀 프로젝트를 하면서 **실력이 많이 상승한 것이 느껴져서 자신감이 생겼습니다.**

지금 생각해 보면 어렵다고 생각이 들때마다 강사님이 말씀해 주셨던 '어려운 게 아니라 안 익숙한 것이다', '어려운 만큼 돈을 버는 것이다 말이 떠올라서 열심히 벼勁다고 생각이 됩니다. 앞으로 진행할 프로젝트에서는 이번 프로젝트를 진행하면서 얻었던 경험, 지식 등을 이용하여 좀 더 노력하게 개발할 것입니다.

팀 프로젝트를 경험한 첫 번째 프로젝트, 결과물이 정상작동 하는 걸 보니 기분이 좋습니다.

후기 문제해결

프로젝트를 시작할 당시, 제가 지금까지 쌓아온 지식을 활용해 손쉽게 진행할 수 있을 것이라는 자신감으로 충만했습니다.

그러나 프로젝트 진행 중 어려움에 부딪히면서 두 가지 중요한 깨달음을 얻게 되었습니다.

우선 첫 번째로, **유효성 검사의 중요성에 대한 깨달음**입니다. 제가 맡은 부분, 즉 공지사항 모듈을 구현하는 것은 수월하게 마무리했다고 생각하고 다른 부분에 눈을 돌렸습니다. 그러나 프로젝트 완료 후 공지사항 부분에서 수정을 시도하니 기존 파일이 사라진다는 심각한 오류가 발생했습니다. 이로부터 기능 구현 뿐만 아니라 해당 기능이 예상대로 작동하는지에 대한 **유효성 검사를 철저히 해야 한다는 교훈**을 얻었습니다. 또한, 내 시각과 타인 시각의 차이를 이해하는 것이 팀 프로젝트에서 효과적인 협업을 위해 필수적임을 깨달았고, 유효성 검사는 혼자 수행하는 것보다 다수의 참여가 품질을 높일 수 있다는 인식을 얻었습니다.

두 번째로, **협업의 중요성을 체감**했습니다. 제가 다음으로 맡은 부분에서 여러 문제가 발생하며 어려움에 처했습니다. 그중 가장 큰 문제는 정규직 개발자의 정보 수정 부분이었습니다. 초기에는 오류 없이 완벽하게 동작한다고 생각했지만 여러차례 테스트를 거쳐보니 가끔은 작동하고 가끔은 그렇지 않았습니다. 이 문제에 대한 해결책으로 메소드 실행 순서를 조정하려 노력했지만 실패했습니다. 개인적인 성격상 맡은 일은 스스로 해결해야 한다는 강한 신념이 있어서 2일간 혼자서 문제를 해결하려 했지만 결국 수정하는데 실패했습니다. 그러나 팀원들은 이 문제가 트랜잭션이 제대로 작동하지 않는 현상이라는 것을 파악하고 각 서비스 메소드에 개별적으로 걸린 트랜잭션을 하나로 통합함으로써 문제를 해결해 주었습니다.

이를 통해 협업의 중요성을 몸소 체험하며 문제를 함께 해결하다 보니 트러블 없이 프로젝트를 성공적으로 완료할 수 있었습니다.

마지막으로, 팀장과 팀원들에게 감사의 말씀을 전하고 싶습니다. 팀원들의 도움과 협력 덕분에 어려움 속에서도 프로젝트를 성공적으로 마무리할 수 있었습니다.

이 **프로젝트를 통해 개인적으로 성장할 수 있는 기회**를 얻었고, 더 나은 협업과 개발의 길을 열어갈 수 있게 되었습니다.

앞으로도 지속적인 발전을 위해 노력하고자 합니다.



“실력향상에 협업은 데!”

후기 문제해결

프로젝트를 통해 팀원들과의 회의를 통해 업무를 분담하고 협력하여 프로젝트를 수행하는 과정에서, 학원에서 이론적으로만 배웠던 지식을 프로젝트에서 실제로 활용하는 좋은 경험을 쌓게 되었습니다. 프로젝트 초기에는 어떻게 시작해야 할지에 대한 불안감과 함께 막연한 부분이 많았습니다. 그러나 팀원들과의 소통과 회의를 통해 목표를 명확히 하고 각자의 강점을 살려 효과적으로 업무를 나누는 방법을 찾을 수 있었습니다.



“오류 찾기는 껌!”

이론적인 지식을 실무에 적용하는 것은 쉽지 않은 일이었지만, 프로젝트에서는 실제 상황에서 발생하는 문제에 직면하며 그동안의 학습을 어떻게 활용할지에 대한 훌륭한 기회가 되었습니다. 프로젝트에서는 이론을 실제로 적용하면서 그 차이를 몸소 느낄 수 있었습니다. 특히나 각 팀원이 가진 다양한 전문성을 살려 문제에 접근하고 해결하는 과정에서, 팀 전체의 역량이 크게 향상되었습니다.

프로젝트를 통해 현업에 대한 불안감과 두려움을 조금이나마 줄일 수 있었습니다. 프로젝트 수행 중에는 여러 어려움에 부딪히기도 했지만, 그 과정에서 발생한 오류들을 찾아내고 수정하는 경험을 통해 실력이 크게 향상되었다는 느낌을 받았습니다. 이는 앞으로의 도전에 대한 대비력을 높이는 데에 큰 도움이 되었습니다. 프로젝트의 성공은 팀원들 각자의 노력과 협력 덕분에 이루어진 것이었습니다. 팀장과 팀원들에게 감사의 인사를 전하고 싶습니다. 각자의 역할과 노력이 조화롭게 어우러져 만족스러운 결과물을 얻을 수 있었습니다. 이러한 팀워크는 향후의 프로젝트에서도 중요한 역할을 할 것으로 기대됩니다. 프로젝트를 통해 얻은 경험은 앞으로의 경력 발전과 프로젝트 수행에서 큰 자산으로 작용할 것으로 믿습니다. 이를 토대로 더 큰 도전에 안정적으로 대응하고, 지속적인 성장을 이루어 나가겠습니다. 프로젝트를 통해 얻은 경험과 지식은 미래의 여러 프로젝트와 현업에서 빛을 발할 것입니다.

Q & A •