

이력서



홍성혁 남 1993년, 30세

휴대폰 | 010-4290-8654 Email | markhong93@naver.com

전화번호 | 02-455-8654

주소 | 서울 광진구 구의3동

학력	경력	인턴·대외활동 / 해외경험	자격증 / 어학
수원대학교 대학교(4년) 졸업	천재교과서 재직 중 총 1년 2개월	-	-

학력

2014. 03 ~ 2020. 02 **수원대학교** 컴퓨터 소프트웨어
졸업 졸업논문 | 한이음 프로젝트 [학점 연계] - 부동산 빅데이터 스마트 챗봇

경력 총 1년 2개월

2021. 01 ~ 재직중 **천재교과서** IT부서 영업개발 1팀 사원 팀원

CRM, MGM 페이지 운영 및 기능 추가 개발

연봉 | 3,100만원

주요직무 | MS-SQL, asp

교육

2020. 04 ~ 2020. 12 **JAVA 프레임기반 풀스택 양성** 코리아 아이티 아카데미

코리아 IT 아카데미

JAVA : 기본

Web : JS, JQuery

Spring : 팀 협업 프로젝트 (RESTful api 진행)

자기소개서

개요

신입으로 지원하는 이유는, 실무에서 도커, Node, Redis 등을 백엔드로써 다뤄보지 않았기 때문에, 새로이 시작한다는 마음입니다.

본래 치기공학과에 재학을 하였으나, 단순한 손으로 만들어내는 것이 아닌, 논리적 과정을 통해 사람에게 도움을 주는 작품들을 창조하는 프로그래밍이 멋있어 보여, 반수를 통해 컴퓨터학과로 다시 재학하게 되었습니다. 전공 과정에서 C와 C#을 다룰 줄 알게 되고, 리눅스도 간단히 배웠습니다. 제 학교의 컴퓨터학과는 학부생 랩실을 따로 뒀었는데, 그곳에 들어가 파이썬을 배우고 파이썬과 openCV를 통해 간단한 얼굴 표정 분석기도 만들어 보았습니다.

졸업작품으로는 한이음 이라는 멘토-멘티 국가 프로젝트 사업에서 진행하는 스마트 부동산 챗봇 프로젝트에 참여하였습니다. 해당 프로젝트의 기획 목적은, 실거래가를 제공하는 기존의 서비스에 더하여 언론기사의 단어를 통해 긍정도 데이터를 가공하고, 가격 예측을 하여 부동산 거래가 변동을 알아 보는 취지를 가지고 있었습니다. AWS에 서버를 Flask로 구축하고, 구글에서 제공하는 Dialog Flow로 챗봇을 동작시키고, 웹훅을 통해, DB에 저장한 부동산 실거래가 공공 API를 ARIMA 공식으로 정보로 가공, 예측 가격을 보여주고, 기사의 단어를 KoNLPy를 통해 형태소를 분석, 형용사의 긍정 부정도 데이터를 제공해주는 작업을 진행했습니다. 다만 이때는 경험과 지식이 턱없이 부족하여, 이미 있는 공개된 코드를 가져다 쓰는 수준에 불과했습니다.

이런 부족함을 배우고 개발자가 되고자, 국비 지원 교육을 통해 자바와 웹 기술을 배우고자 교육기관을 찾았고, 보다 실무에 가까운 JAVA를 배워 객체 지향에 대한 기본을 다지고, 웹 기술들을 배워 뷰 페이지를 작성 가능하게 되었습니다.

또한 Oracle DB의 사용에 능숙해지고, 이를 바탕으로 JAVA Swing을 이용한 테블릿 연동 포스기 프로젝트를 진행하였습니다.

본래 웹에 대해 기반이 약했지만, 과정 통해 기본을 확실하게 다졌습니다.

그 후, 천재교과서 IT 영업 개발팀에 들어갔습니다. 고객 회원과, 교사 회원을 위한 영업, 마케팅의 서비스와 통계를 다루는, CRM, MGM 페이지 운영 및 기능 추가 개발 등을 하였습니다. 해당 서비스는 ASP와 SQL Server로 이루어져있었습니다. 서비스는 레거시 패턴으로 서블릿 방식의 서버이며, 비즈니스 로직은 전부 데이터베이스 단에서 프로시저로 다루어지고 있었습니다. 처음 접하는 프로그램 체계였지만, 스프링과 JSP 학습의 경험에 빚대어, 해당 체계를 빠르게 학습하고, 업무를 시작 할 수 있었습니다.

자바 Swing으로 구현한, 테블릿 연동 포스기

자바의 Swing을 통해 GUI로 구현한 테블릿 연동 포스기 프로젝트는 기본적으로 헌팅포차의 시스템을 가져왔습니다. 당시, 코로나와 클럽의 폐쇄로 헌팅포차가 매우 인기를 끌었습니다. 이런 와중에 테이블 테블릿을 통한 메뉴 주문이 눈에 들어왔습니다.

제 팀 프로젝트의 고객은 매장의 사장님들입니다. 매장은 테블릿을 이용한 메뉴 주문 시스템을 이용하여 많은 이득을 거둘 수 있습니다. 첫째로, 주문량의 증가입니다. 기존 직원이 메뉴 주문을 받을 때에는 직원의 숫자에 한계가 있기 때문에, 손님의 주문시에 직원이 반응하기까지의 대기시간이 주어질 수 밖에 없고, 이는 주문량의 감소 이어집니다. 하지만 테이블 테블릿 메뉴판을 통해 주문을 하게 된다면, 이 문제점을 해소할 수 있고 이는 주문량 증가로 이어지게 됩니다. 두번째로, 주문 누락이나 오류가 사라집니다. 테이블 테블릿을 통한 주문으로 주방으로 바로 주문이 들어가기 때문에 직원의 실수를 방지 가능합니다. 셋째로, 증가하는 인건비를 감소시킬 수 있습니다. 추가로 게임 기능을 넣어, 타 테이블과 테블릿을 통해 주문 대기시간 동안 게임을 통해 소통을 할 수 있게 하여 흥미도를 높였습니다.

전 프로젝트의 서버를 제작하고, 각각 서버와의 통신 연결을 담당할 클라이언트를 만들었습니다. 어떤 프레임워크도 사용하지 않았고, 어떤 통신이 요구될지 몰랐기 때문에, 정확한 전송을 위해 TCP 방식을 채택하고, 모든 객체를 전송 가능하도록 하기 위해, 메시지를 클래스로 작성하여 전송 양식을 일반화 하고 팀원들에게 배포하였습니다. 또한 각 클라이언트는 다형성을 사용하여, 연결될 각 테이블과 주방, 카운터에서 인터페이스를 구현하여 접근 가능하게 하였습니다.

초반 기획 단계에서 활발한 브레인 스토밍을 통해 다양한 아이디어들을 제시했습니다. 팀의 역량에 맞는 구현 가능한 아이디어를 골라내고, 기획을 확정하는데 난이도가 높아서 1주일 넘게 걸렸습니다.

리더는 높은 달성 목표를 제시하고, 모범을 보여야 한다고 합니다. 프로젝트 팀에서 저 혼자 전공자였기 때문에, 제가 기능 구현에 대해 목표를 정하고, 역할 분담을 해주었습니다. 진행중에 자신과 팀의 역량에 대한 정확한 판단을 하지 못하고 너무 높은 목표를 제시하여, 계획이 틀어지고, 기한에 문제가 생기고, 팀원 간의 갈등이 생겼습니다. 역량에 대한 재평가를 하고, 각자의 목표를 수정했으며, 팀원의 화해를 이루고 협업하여 프로젝트를 잘 마무리 할 수 있었습니다.

프로젝트를 진행한 뒤 저는 두가지 큰 깨달음을 얻었습니다. 첫째는 경험에 대한 겸손입니다.

역량과 견적에 대한 경험이 너무나 부족하다는 것을 알았습니다. 안으로는 자신의 능력을 과대평가했고, 밖으로는 아무리 높은 목표라도 상황을 잘 이 끌어 나갈 수 있다고 생각했습니다. 아직 경험하지 못한 것에 대해서는 함부로 판단하지 말고, 어떤 상황에도 빠르게 대처할 수 있는 기술 스택을 꾸준히 쌓는 것이 필수라는 것을 알았습니다.

두번째는 작업에 대한 자신감입니다. 이번 프로젝트를 통해, 어떤 문제를 받아도 작업해 낼 수 있다는 자신감이 생겼습니다. 프로젝트의 전반적인 부분을 다루게 되니, 어느 부분에서 진행이 느려지고, 자잘한 요구사항이 드러나는지에 대한 탐구를 할 수 있는 경험이 되었습니다.

JSP로 구현한, 회원제 게시글 커뮤니티 페이지

톰캣, JSP를 이용하여, 기본적인 기능들로 다이내믹 웹 프로젝트로 만들었습니다. 최근 메타버스라는 말이 조금씩 떠오르고 있습니다. 메타버스 VR 공간에서 사람들과 만난다는 것이지만, 따지고 보면 커뮤니티 사이트가 그 시작이라고 할 수 있을 것 같습니다. 그에 착안하여 가장 기본기를 확실히 익히기 위하여, CRUD를 모두 갖춘 게시글 커뮤니티 페이지를 만들어보자고 팀에게 제안했습니다.

우리 게시글 커뮤니티 페이지는 글의 내용에 제한이 없습니다. 글의 크기가 책 한권이어도 괜찮습니다. 저장 방식의 차이입니다.

여러 게시글 사이트를 돌아보다가, 게시글을 사진을 저장하듯이 파일로 저장하고, DB로 경로를 관리하면 어떨까 하는 생각이 들었습니다. 게시글 에디터는 글을 HTML 문자열로 저장합니다. DB의 CLOB 자료형 처럼, 직접 관리할 수 있게 구현한 것입니다.

또한 AJAX를 활용하여, 페이지의 주소를 옮기지 않고, 페이지의 내용만 바뀌는 SPA를 지향하여, 무한 스크롤 기능이나, 페이지네이션 기능을 추가했습니다. 파일로 저장한 게시글의 내용을 파싱하여 리스트로 보여주기 위한 중간 매개체 프로세스를 작성하고, 검색기능이 가능하게 했습니다. 기존의 프로젝트 경험들을 반면교사 삼아, 프로젝트 규모를 크게 잡지 않은 대신, 새로운 아이디어, 새로운 기능들을 접목시키고, 활용하는데 집중했습니다.

다만, 이번엔 팀원들의 역량에 비해 목표치를 약간 낮게 계획하여, 프로젝트의 규모가 기간에 비해 작았던 것 같습니다. 또한 프론트 엔드 쪽에 능숙하지 못하여, 뷰의 디자인 결과가 조금 아쉬웠습니다.

프로젝트 기획에 있어, 프로젝트 규모를 단계별로 늘리는 것이 어떨까 생각해보게 되었습니다. 계획은 기간에 맞춰 많은 기능들을 단순하게 작성하되, 작업이 빠르게 진행되어 기간이 남게 된다면, 내부 리팩토링을 거쳐 더 좋게 최적화를 해야겠다고 생각했습니다.

또한 프론트 단의 디자인을 구성할 때, 시간을 단축 시키고 완성도를 높이기 위하여, 내가 모든걸 작성하려 하지 말고, 기존의 부트 스트랩 템플릿과 여러가지 API를 활용하는게 합리적이라는 생각을 갖게 되었습니다.

첫 웹 프로젝트는 다시한번 제게 할 수 있는것, 효율적인 것에 대해 생각하게 되는 기회가 되었습니다.
협업이란 내가 할 수 있는 것이 있고, 해야할 것이 있고, 하기 힘든 것이 있다는 것에 대해 깨닫는 기회였습니다.

Spring을 이용한, 도로교통공사 api 활용 커뮤니티 페이지

Spring, MyBatis, 오라클DB를 이용하여 스프링 레거시 프로젝트로 만들었습니다. 이전의 게시물 커뮤니티 페이지를 작성한 경험을 더해, 외부 API 데이터를 사용하여 의미있는 서비스를 제공하는 프로젝트를 만드는데 동의했습니다. 여러 데이터를 조회하다, 휴게소, 주유소, 음식메뉴라는 통합 서비스를 커뮤니티 서비스와 엮어 제공하는 것은 없다는 것을 알게 되었습니다.

이번 프로젝트의 목적은 여러가지 스프링의 장점들과, 디자인 패턴들을 적용시켜보자였습니다. 단순한 Monolithic Architecture 이지만, AOP를 이용하고, RESTful API로 만들어 최대한 Single page Application을 지향하고, MVC 패턴을 활용하기로 하였습니다. 또한 이번에는 팀원들에게 적극적으로 다형성과 매핑의 유용함을 주장하여, DAO, DTO, VO를 체계화하는데 힘썼습니다.

우리 페이지는 기존 게시물의 좋아요, 댓글 등의 기능들을 모두 갖춘채, 일정 시간, 혹은 관리자의 관리 하에 데이터 최신화를 자동으로 해주는 페이지가 되었습니다.

이번 프로젝트의 아쉬운 점은 Spring boot가 아니기 때문에 Junit, JPA 같은 것의 활용이 제한적이었던 것입니다.
빠르게 Spring boot를 공부하고, React를 공부하여, 웹페이지의 기본을 마스터해야겠다고 다짐했습니다.
또, 프로젝트 중 한 친구가 포기해서 제가 두 파트를 맡게 되었습니다. 이전 프로젝트 때에도 그 친구가 안하려고 해서 힘들었지만 이끌고 해결했지만, 이번 과정에서는 코로나로 인해 비대면으로 프로젝트를 진행하다보니, 그 친구가 손놓으니 방법이 없었습니다. 다음에 협업 과정에서 이런 상황이 생긴다면, 어떻게 더 좋은 방향으로 해결해야할지 고민점을 갖게 되었습니다.

밀크T유아 프로젝트 - 교사 메시지 관리 페이지

페이지 작성과, 비즈니스 로직이 담긴 프로시저 작성.
중학 CRM 메시지 기존 페이지는, 템플릿의 JS에 스크립트릿이 파편으로 섞여있어, 로직을 이해 할 수 없었습니다.. 기획서도 찾아볼 수가 없었습니다.
정책을 물어볼 사람이 없어,
한줄 한줄 로직을 분석해서, 스크립트릿을 걷어내고, JS 코드에 FP 원칙을 지향하여 작성하였습니다.

기존 중학 메시지 관리 서비스는 씨드 파티(CJ_Msg)를 이용하고 있는데, 디비 링크 방식으로 인터페이스가 처리되고 있었습니다.
또한 기존 비즈니스 로직이 담긴 프로시저에는 여러 메세지를 보낼 경우, 로직 처리를 위해 커서를 이용하고 있었습니다. 쿼리들의 로직을 해체하고, 분석해서, 조인 연산 등의 집합적 연산 방향을 지향하고, 정말 필요한 곳에서만 커서를 사용하도록 프로시저들을 튜닝하였습니다.

통합테스트 직전에 유아 회원 테이블의 아이디 체계가 급변했습니다. 기존 중학 방식은, GET 방식으로 회원 아이디들을 넘겨주었는데, 체계가 바뀌며, 해당 방식을 사용할 경우, 장문의 시스템 아이디들이 넘어오며 문제가 생겼다. URL 길이가 초과하여, 에러가 생겼습니다. 더불어, 중복 체크도 안하고 있었습니다. ES6+ 의 문법을 활용하여, 중복 체크 및 선택적 GET , POST 리퀘스트 방식으로 리팩터링 하였습니다.

나중에 유아 프로젝트를 진행하며 발견한 문제점들은, 역으로 중학 메시지 페이지에 반영하였습니다. ES6+ 문법의 경우, 중학 CRM은 IE 환경을 지원해야했기 때문에, (바벨도 사용하지 않아) ES6+로 짠 코드를, 이하 JS 버전 환경에 맞게 작성하는 경험도 했습니다.

중학 mcrm, mmgt 서비스 기능 추가 및 개선

- 유아초,중고 영업관리 시스템을 하나로 합치는 HUB 프로젝트가 진행되며,
유아, 초, 중고 CRM들의 여러 기능들에 데이터가 추가되는 작업들이 있었습니다.

1. 물류 관리 페이지 기능 추가 및 관리

물류 페이지 또한 써드파티 WMS와 연결되어 있습니다. 해당 시스템은 API를 통해 연결되고 있습니다. 테이블에 새로운 데이터 컬럼 추가 적재를 요청 받았습니다. 해당 시스템은 모델을 사용하고 있었는데, 발송과 회수가 같은 모델을 이용하였습니다.

새로운 데이터 컬럼이 추가되어야 하는 부분이, 모델 안의 모델이었습니다.

인터페이스를 추가하여, 데코레이터 패턴 및 상태 패턴을 이용하여, 발송과 회수 상태에 따라 모델을 관리하도록 반영하였습니다.

2. 교사(CRM 회원) 검색 기능

교사에게 권한을 주는 페이지에 권한 복사 기능이 있었습니다.

문제는 복사 주체자-대상자를 선택하기 위해서, 전체 교사 아이디가 나왔습니다.. 권한 복사하기가 너무 힘들다고, 기능 수정 요청이 들어왔습니다.

간단히 input을 두개 만들어서, 쓰로틀링을 통해 ajax 요청 횟수를 줄이고, 교사 아이디를 필터링 하는 기능을 추가했습니다. (라이브러리 없이 es5로 작성하여, 쓰로틀링을 직접 구현해야 했습니다.)

3. 교사(CRM 회원) 직급 체계 추가

중고 CRM의 교사 직급 체계는, 직급 코드와 상위 교사 아이디로 관리되고 있었습니다.

초등의 경우, 조직-팀-파트 데이터로 관리 되고 있었습니다. 허브의 통합 데이터 관리를 위해, 조직-팀-파트 데이터 입력 기능 추가를 요청받았습니다.

사용자 등록, 수정 페이지에 조직-팀-파트 컬럼이 추가되며, UX를 고려한 스크립트 및 페이지 작성을 했습니다. 스크립트는 JQuery를 지양하고, 바닐라를 지향했으며, ASP의 C# 코드 또한 클린코드를 지향하여 작성했습니다.

추가로, 기존 교사들에 대한, 만 건 이상의 데이터 처리와, 테이블 수정 등 DBA와의 협의를 통해 작업되었습니다.

4. 대량 메시지 전송

이벤트 기간에는 몇 만 건 조금 넘게, 한번에 문자를 발송해야 하는 일이 있습니다.

하지만, 기존 프로시저를 사용하여 디비 링크드 테이블에 적재하게 될 경우, 커서를 사용하기에, 디비에 무리가 됩니다.

때문에, 커서가 사용되고 있는 프로시저를 낱말이 분석해서, 임시 테이블과 조인 연산을 통해, 다량의 데이터를 처리하는 로직을 작성하여, 데이터를 적재하였습니다.

5. 초등에서 중고등 서비스로 회원이 이관되는 것과 교사의 관리로 인해, 학년 승급은 자동으로 처리가 안되게 되어있습니다. 때문에 수동으로 데이터를 업데이트 처리해야하는데, 모든 학생 회원의 값을 처리해야 합니다. 몇 십만건의 데이터를 위해, 쿼리를 작성하고, DBA와 협업하여, 데이터베이스를 모니터링하며, 반영하는 경험을 했습니다.

5. 그 밖의 요청으로, 영업 활동을 위해, 기타 대량의 데이터를 요청 및 강제 반영하는 경우가 있었는데, 이전 쿼리들의 로직을 분석하여 반 모듈화 시켜서, 필요한 값만 넣으면 처리가 가능하도록 작업하였습니다.

지난 1년 동안, 회사와 병행된, 다양한 공부

- 디자인 패턴과 클린 코드

학원에 다니며 항상 제 코드의 품질에 대해서 의심을 했습니다. 코드를 작성하다보면, 어느순간 더러워져가는 제 코드를 보며, 더 나은 코드를 작성하기 위한 기본이 필요하다고 느꼈습니다.

때문에 당시 입사 전, 디자인 패턴을 공부했습니다. 또한 클린 코드에 대해서도 공부했습니다.

- ASP와 SQL Server

업무를 위해, 총 이주일 동안 열심히 공부하였습니다. 그 이후로도 마이크로소프트에서 제공해주는 공식 문서를 참조합니다.

- 모던 자바스크립트 deep dive 스터디에 참여하여, JS의 기반을 다졌습니다.

실행 컨텍스트 개념, 프로미스 개념, this, 화살표 함수, 프로토타입 개념, 이벤트 루프, 쓰로틀링과 디바운스 등에 대해 알게 되었으며, 자세히 토의하고, 선배 개발자들의 다양한 실무 경험담을 들을 수 있었습니다. 해당 스터디를 통해, 검색 방법의 중요성을 알게 되었습니다.

이를 통해 자바스크립트의 매력에 푹 빠졌습니다.

- 쿼리 튜닝

인덱스와 조인 연산의 다양한 팁들에 대해 알게 되었습니다. 데이터베이스에 대한 자신감이 생겼으며, 동시에 방대하고 먼 길에 대해 알게되어, 겸손을 가지게 되었습니다.

- 노드 익스프레스와 Vuejs, 도커, 라즈베리파이

공식 문서와 기초 책으로 공부하였습니다. 다만 당시에는 이것이 주가 아니었습니다.

당시 입사 후, 어느정도 기간이 지나고, IoT 환경 구축에 푹 빠져있었습니다.

상품들에 있어, 각자 지그비, 블루투스, 크롬캐스트, 에어플레이 등 통신 프로토콜이 달라 통합이 되지 않았으며, 어플도 제각각이었습니다. 때문에 스마트싱스와 구글 홈, 카카오톡 홈, 애플 홈 API 들을 연결해주는 프로그램을 만들기 위해 각 기술들을 공부하였습니다.

MSA로 각각의 API 라이브러리 들을 서비스로 올려두고, 메시지 큐, 외부와 API로써 연결되는 입구 부분 라우터 서비스, 그리고 각 서비스들을 이어 붙여 줄 프록시 모델과, 그런 프록시를 관리할 비지터 패턴으로 작성되는 설정 서비스를 구상하고 만들어보았습니다.

다만, 각 사의 API 가 제대로 지원되질 않고, 자료 수집이 난해하여, 멈췄습니다.

조금 더 공부하고, IoT 시장이 더 커지고, 지원이 다양해지면, 다시 재개해 볼 생각입니다.

- 도커 재학습

하지만, 실무에서는 도커 스웜을 쓰질 않아, 공부해도 도커에 익숙하지 않았습니다. 때문에 쿠버네티스까지는 공부하지 못했습니다. 도커 위에서 마인 크래프트 서버를 돌리고 친구들에게 운영하는 사이드 경험을 해보았습니다.

- 또한 최근에 TypeScript 스테디를 진행중 입니다.

장단점

1. 저는 분석적이고, 디자인 능력이 높습니다.

저에게 일이 주어진다면, 바로 행동하지 않고 계획, 디자인을 합니다.

회사 월급 받고 처음 돈 쓴 곳이 아이패드를 사는 일이었습니다. 아이패드의 굿노트 앱에는 제가 업무를 받으면 고심하는 분석과 고민의 흔적들로 가득 합니다. 이것은 프로그래밍을 할 때 보기 좋은 빌드 업으로 드러납니다. 요구사항을 단순히 분석할 뿐만 아니라, 클래스와 함수 하나하나 모듈화하고, 일반화하여, 코드의 재사용성을 높이고, 유지보수의 능률을 높입니다.

그것은 프로젝트에서 제가 맡은 역할에서도 드러났습니다.

제가 잔 코드들은 프로젝트 초반에 기능이 적더라도 크기가 크고, 완성까지 시간이 꽤나 걸렸습니다. 하지만 프로젝트 중후반에 들어서자, 어떤 자잘한 요구사항들이 더 드러나도, 기능을 일반화 시키고, 구조화 시켰기 때문에, 기능 추가 등의 유지보수가 쉽고, 작업 속도가 빨랐으며, 보기가 쉬우니 에러가 발생하질 않았습니다. 설사 에러가 발생해도 어디서 발생했는지 금방 그 위치를 찾아낼 수 있었습니다.

(물론, 일년이 지난 지금, 프로젝트 당시에는 클린 코드 패턴을 잘 모르고 있었기 때문에, 그 코드를 보면 토가 나오긴 합니다.)

2. 또한 저는 문서화 광입니다.

저는 보조기억장치가 아닙니다. 프로그래머란, 컴퓨터로 치면 CPU와 주메모리장치입니다. 맨날 쓰는것은 캐시해두겠지만, 대부분은 지식 키워드를 기억해두고, 검색하여 사용하는 해시 기능이 막강한 프로그래머가 좋은 프로그래머라고 생각합니다. 회사는 문서 작업이 빈약했습니다. (SVN을 씁니다.) 항상 요청이 있을때마다, 백업과 수정파일, 이력파일을 남겨서, 제가 뭘 했는지 남깁니다.

또한 매우 친절하여서, 신규 입사자가 들어와서, 혹은 팀원이, 웬만해서는 제게 물어보기 앞서 제 작업 이력을 확인하고 처리됩니다.

제 이런 신중함과 작업 능력은 어디에도 적용 가능하며, 이런 능력을 통해 회사에서 새로운 프로젝트를 작업할 때, 유지보수 작업이 필요할 때, 쓰이는 인재가 되었습니다.

3. 저는 분석적이고 일반화하는 것을 좋아하기 때문에, 이해를 중요시합니다.

자신이 작성하는 것은, 코드 한줄, 설정 한줄도 모르고 쓰면 안된다고 생각합니다. (물론 불가피한 사정상 알지 못하고 쓰는 경우가 있습니다. 그러면 매우 괴로워 합니다.)

또한 해당 방식은, 새로운 지식의 크기가 작다면, 습득 속도가 빠른 편이지만, 크기가 클수록 습득 속도가 느린 편입니다. 이것은 새로운 지식을 이해하고 머릿속에서 일반화하는 과정이기 때문에, 초반엔 느려도, 나중에는 오히려 응용속도나, 비슷한 지식에 대한 학습속도가 빠릅니다.

그 예로, C#에 대한 기본을 끝까지 배웠기 때문에, 그 후 Python이나, JAVA 기본 공부에서 굉장히 빠르고, 깊게 기본기를 다룰 수 있었습니다.

또한 처음 보게 된 ASP를 쓰는 회사의 방식도, 1~2주 만에 익힐 수 있었습니다.

이렇듯이 항상 꾸준한 공부를 통해, 지식을 쌓아 어떤 지식을 접하게 되도, 빠르게 학습 가능하고 응용 가능한 상태가 될 수 있게 노력합니다.

입사 후 포부

훌륭한 프로그래머의 조건은 무엇일까요?

oop에서 제일 중요한 것이 추상화인 것처럼. 어디에서도 쓰일 수 있는 기본을 갖추는 것은 프로 개발자라면 당연한 것이며.

또한 주니어 개발자라면, 클린 코드 패턴에 대해 얼마나 아는지가 상당한 비중을 차지한다고 생각합니다.

패턴에 대한 지식을 가지고 있는지 차이는, 단순히 코드의 질에서 벗어나, 프로그램을 효율적이고, 지속적이며, 아름다워지게 만든다고 생각합니다.

단순히 oop가 어떻고, fp가 어떻고 하는 것에 그치지 않고, 패턴을 익히는 것이 중요하다고 생각합니다.

때문에 C#, Python, Java, Javascript 등 다양한 언어들의 기본을 쌓았으며, 여러 프로젝트들을 진행하며 다졌습니다.

또한 나름 규모있는 회사에서 레거시 시스템을 다뤄봤습니다.

이젠 MSA를 다뤄보고 싶습니다. FP와 OOP를 실무에서 다뤄보고 싶습니다. 실시간으로 최신화되는 기술을 바라보기만 할 뿐 아니라, 회사와 함께 이것을 반영이 가능할지, 고심에 참여하고 싶습니다.

어려운 일을 맡게 된다는 것은, 그 일을 해결할 수 있는 사람이 된다는 것을 의미합니다.
그것은, 성장의 가능성을 의미합니다. 제가 맡은 일들은 때론 도무지 끝이 보이질 않았습니니다.
하지만 저는 목표를 제시하고, 팀원들과 같이 나아갔고,
회사에서는 팀원으로써 제 일을 철저히 이루고, 기대 이상을 냈습니다.
결과를 얻어내 왔으며, 성장해왔습니다.
제가 할 수 있는 것을 하겠습니다.

포트폴리오

포트폴리오	https://github.com/markhong93/portfolio
-------	---

희망근무조건

고용형태	정규직
희망근무지	서울전지역, 경기 성남시 분당구
희망연봉	면접 후 결정
지원분야	직무 웹프로그래머 > 웹프로그래머

위의 모든 기재사항은 사실과 다름없음을 확인합니다.

작성자 : 홍성혁

이 이력서는 2022년 02월 12일 (토)에 최종 업데이트된 이력서 입니다.
위조된 문서를 등록하여 취업활동에 이용 시 법적 책임을 지게 될 수 있습니다.
잡코리아(유)는 구직자가 등록 한 문서에 대해 보증하거나 별도의 책임을 지지 않으며
첨부된 문서를 신뢰하여 발생한 법적 분쟁에 책임을 지지 않습니다.
또한 구인/구직 목적 외 다른 목적으로 이용시 이력서 삭제 혹은 비공개 조치가 될 수 있습니다.