

스스로 디버깅하고 발전하는 신입 백엔드 개발자 박윤택입니다.



박윤택 1996년 (27세/만 25세) | 남 | 재직중

✉ dbxsortime@naver.com

☎ 010-8998-0372

☎ 010-8998-0372

🏠 (08110) 서울 양천구 남부순환로

📌 **핵심역량** ▪ 서버 고도화 초기 진행 중

- 실제 사용되고있는 관리자 웹을 유지/보수 하며 요구사항을 파악하고 개발하는 업무
- 로컬/개발/테스트/운영 서버의 단계별 개발, 테스트 및 배포
- SVN, zenkins, hiware의 사용으로 협업의 이해

학력사항	경력사항	희망연봉	희망근무지/근무형태	포트폴리오
고등학교 졸업	경력 3개월	회사내규에 따름	서울전체 정규직	총 2건

학력 최종학력 | **고등학교** 졸업

재학기간	구분	학교명(소재지)	전공	학점
2012.03 ~ 2015.02	졸업	백암고등학교		-

경력 총 3개월

근무기간	회사명	부서/직급/직책	지역	연봉
2021.10 ~ 재직중	스마트로	개발실 / 임시직/프리랜서	서울	-
	📌 담당업무 서버 고도화프로젝트 및 관리자 웹 유지보수 개발			

대외활동

기간	구분	기관/장소	내용
2021.01 ~ 2021.07	교육이수내역	구트아카데미학원	Java기반 빅데이터 웹프로그래머 양성과정
2021.11 ~ 2022.01	교육이수내역	학점은행 이수 중	학점은행 컴퓨터공학과정 교육과정 이수중

보유기술


보유기술명/수준/상세내용
Zeplin
GitHub

Oracle SQL
Database
Python
Spring Framework
CSS3
HTML5
JavaScript
Java

취업우대사항

병역대상	군필 2016.02 ~ 2017.11 육군 병장 만기제대
------	---

포트폴리오/기타문서

파일 구분	파일명
포트폴리오	 ppt.pdf 3.4MB https://dbsxortime.github.io/
이력서	http://github.com/dbsxortime

자기소개서

[성장과정]

#디버깅 #나를디버깅

살면서 실수들이 있었습니다. 결과에서 문제가 생기기도 하고, 과정에서 찾게 된 문제들이 있습니다. 그런 문제들이 생길 때 어디서 문제가 생겼는지 파악하거나, 기획 단계에서 문제점을 파악하고 미리 보완한 경험들, 그리고 그 데이터를 가지고 같은 문제를 다시 만들지 않는 습관이 쌓여 이런 경험을 하지 못한 사람들보다 한발 앞설 수 있는 좋은 습관이 있습니다.

코드를 짜다 보면 결과에서 마주치는 수많은 오류가 있습니다. 물론 에러가 나지 않고 한 번에 깔끔한 코드를 작성하는 것도 실력이겠지만, 발생한 오류를 파악하고 처리하는 것 또한 개발자의 역량에 있습니다. 살면서 그래왔듯 문제를 해결하기 위해 계속해서 던지는 '왜'라는 질문 덕분에 누구보다 빠르게 과정과 결과를 이해하고, 디버깅할 수 있게 되었습니다. 또한, 잘못된 결과와 과정에 대해 이해하려 하다보니, 과정에서부터 문제점을 찾을 수 있게 되었고, 점점 오류 없는 결과를 원하는 대로 만들 수 있게 되고 있습니다.

[성격의 장 단점]

#장점 #응용력만렙 #의외로 #노력형인간 #단점 #스스로화가남

처음부터 완벽한 사람이 있을 수 있습니다. 처음 접하는 분야에서 재능을 찾을 수도 있고, 누구보다 완벽하게 일을 하는 사람이 있을 수 있습니다. 저는

단연코 그런 사람은 아닙니다.

오히려 제게 재능이 있다면 천재라는 단어와는 반대로 부족한 부분이 항상 있지만, 어떤 부족함이 있는지 스스로 빠르게 깨닫고 보완할 줄 아는 '노력 하는 방법'에 재능이 있는 사람'입니다. 이 때문에 많은사람들이 제가 어떤 걸 하던지 잘할 수 있는 사람이라고 오해하기도 하는데, 오히려 처음 접하는 개념에 대해서 이해력이 뛰어나지 않습니다.

예전에 한 아이는 2+9가 왜 11인지도 이해하지 못했습니다. 그 아이는 문제와 풀이를 보며 문제를 이해하려 했고, 풀이를 이해하려 했고, 결국 두 자리 개념을 혼자 이해하여 배우지도 않은 두 자리 덧셈 뺄셈을 스스로 할 수 있게 되었습니다. 네 물론 저의 이야기였습니다. '왜 이걸 이 정도로 이해 못하지?' 하는 모습이 있습니다. 2+9를 며칠을 고민하지만 이해한 뒤에는 25+46같은 문제도 스스로 풀 줄 아는 학생이었고, 제가 문제를 보는 방식이었습니다.

'많은 문제를 푸는 노력'이 아닌 '몇 개의 문제를 끝까지 이해하는 노력'을 하는, 사람들과는 다른 노력의 방식으로 다른 사람들이 봤을 때 '처음 해도 잘 하는 사람'이 되지 않았나 합니다. 뒤에서 하는 제 노력을 모르는 사람들에게 가끔 서운하고 억울하기도 하지만, 그만큼 보이지 않게 노력을 잘했다는 뜻일 것입니다.

이런 노력은 포트폴리오 프로젝트때에도 다르지 않았습니다. 웹 게시판 제작 수업할 때 개념이 이해가 가지 않아 누구보다 많은 시간을 이해하는 데 쏟았고, 결국 이해하고 응용하여 포트폴리오 프로젝트로 준비했던 쇼핑몰에서 장바구니와 결제, 결제확인, 공지, 질문 게시판 등 많은 기능에서 스스로 응용하고 설계하여 구현해 내었습니다.

너무 좋은 소린만 적어두었지만, 이론적인 개념에 매달려 이해하려는 모습에도 저 나름의 단점이 있다고 생각합니다. 상위의 태그에 적혀있듯 나 자신에게 화가 많이 있기에 이해하고 넘어가려 하는 모습이 있습니다. 웹 게시판을 공부할 때 예제를 그대로 적은 것 같은데 내 코드만 작동하지 않았고, 세 시간을 매달렸지만 결과가 달라지지 않았고, 늦었기에 그대로 자버렸지만, 새벽에 잠결에 몰마시고 방에 들어가는 길에 내 코드만 작동하지 않은 것이 생각나 화가 나 다시 붙잡고 예제 코드 한 줄 한 줄 뜯어보고 이해하고 다시 내 코드로 구현하고 작동하는 것을 보고 다시 잠이 들었습니다.

물론 이런 것 때문에 다음 수업이나 일정에 지장을 가지거나, 주변 사람들에게 스스로의 화로 인해 영향을 끼치는 아마추어적인 모습을 보인 것입니다. 중요한 것은 저는 화가 있는 사람이었고, 그 화를 자기 계발적인 측면으로 표출하여 긍정적으로 만들 줄 안다는 것. 그리고 계속해서 스스로 점검하고, 부족한 것, 잘하는 것을 이해하려 하고, 스스로 디버깅하며 '단점을 장점으로 승화시킬 수 있는 사람'이라는 것입니다.

[팀 프로젝트 경험담 및 기술 역량]

프로젝트를 설계하고 구현하는 과정에서 화면에 필요한 데이터를 서버에서 불러와 처리해 보내주고, 서버에 필요한 데이터를 화면에서 불러와 처리해 보내주는 팀원들과 계속해서 소통해야 하는 위치에 있었습니다. 끊임없는 소통과 설계, 구현으로 어떻게 데이터가 이동하는지 이해하고, 팀원들에게 요청하고 전달하는 그 과정이 있었습니다.

물론 처음 하는 웹 기능설계였기에 시행착오가 있었습니다. 기능별 작업 순서와 작업 기간책정을 잘못하여 구현 순서가 맞지 않아 나올 수 없는 결과를 도출해내려 한 경험이 있습니다.

쇼핑몰 웹 프로젝트였는데, 구매목록 페이지를 먼저 구현하려 한 실수가 있었습니다. 가상 구매데이터를 입력하고, 불러오려 했었으나, 장바구니, 결제 등의 다른 순차적인 데이터 연계를 생각하지 못하고 결과부터 처리하려 했던 것이 문제였던 것입니다. 순서상 현 작업을 덮어두고 다시 순차적으로 진행이 돼야 했었습니다. 이를 깨달은 후에도 다른 팀원들에게 저의 오류를 설명하고 팀원들과 다시 다른 작업에 들어가야 한다는 부분에서 많은 심적 부담감이 있었으나, 빠르게 잘못을 인정하고, 올바른 일정대로 다시 하는 게 팀에 도움이 된다는 판단을 했습니다. 그 즉시 조장에게 설명하고, 일정을 조정하였고, 다 같이 순차적으로 처리한 결과 일정에 큰 차질없이 프로젝트를 진행할 수 있었습니다.

위와는 다른 문제로 팀원 이탈에 대한 문제가 있었습니다. 인간적인 마찰은 양보와 이해로 해결할 수 있었겠지만, 팀과 개인 간 작업 방식이나 결과물에 대한 스타일이 달랐기에, 서로 공부하고 구현하고 싶었던 게 달랐던 문제였기에 잘 마무리하고 다시 작업에 들어가야 했습니다. 그런데 문제는 6인으로 시작했던 인원이 4명까지 줄었기에 일정 맞추기에 어려움이 생긴다는 것이었습니다. 거기에 저의 오류도 팀원 이탈 직후에 발견되어 일정이 더욱 빠듯해졌습니다. 다들 정신이 없어진 상황에 당장 처리해야 할 문제가 쌓여있고, 쳐내기 급급한 상황이라 느껴져 잠시 작업을 다같이 중단하고, 눈 앞의 처리할 문제만 보는 게 아니라 전체적인 일정 조정을 제한했고, 기존의 '화면', '데이터 처리', '서버'로 선 곳들 나누어 처리하던 업무를 서로 부담을 같이 덜어주며 같이 처리하는 유동적인 방식으로 변경하고, 전체적인 일정 또한 조정하게 되어 조금은 빠듯했지만, 다 같이 힘을 내어 프로젝트를 완료할 수 있었고, 협동심과 전우애를 느낄 수 있었습니다.

아직 학생이기에 이런 경험이 도움이 됐다고 생각합니다. 계속해서 생기는 구조 설계상 오류를 잡아가며 부분 수정을 하거나 설계를 처음부터 다시 하는 과정에서 하나의 기능을 위해 데이터 이동설계를 어떻게 해야 하는지 더 이해할 수 있게 되었던 값진 실수가 있었고, 값진 실수보다 더 아름다운 해결과 발전이 있었습니다.