

송준형 Back End 개발자 이력서



이름: 송준형

생년월일: 1999. 01. 05

전화번호: 010-9520-0105

이메일: ssong9520@kakao.com

포트폴리오 사이트: www.junhyeung.info

학력: 인하대학교 통계학과 / 빅데이터 분석학 졸업 (2021)

자격증

- 정보처리기사 (필기 합격)
- SQL 개발자

저는 높은 적응력과 빠른 습득력을 무기로 빠른 도약이 가능한 개발자입니다.

빠른 러닝 커브와 저의 강점인 프로젝트 흐름에 대한 분석 및 파악 능력으로 새로운 기술 및 환경이 주어지더라도 맡은 업무에 대해 빠르게 시작하고 시간을 확보하여 프로젝트를 제시간에 성공적으로 수행할 수 있는 역량을 가지고 있습니다. 또한 습득한 기술에 대해서 어떻게 다양하게 사용하면 좋을지 항상 생각하여 기술 적용의 확장 및 유연성을 확보하려고 언제나 노력하고 있습니다.

Tech Skills

Back End

Java

Java 언어에 대해서, OOP의 개념에 대해서 공부하고 이해하고 있습니다. 객체 지향적인 코드를 작성하도록 노력하고 있습니다. Java를 주력 언어로서 사용하면서 해시, 스택, 큐 등의 자료구조와 정렬, DFS, BFS와 같은 알고리즘에 대해 학습하기 위해 사용했습니다. 학습한 내용을 바탕으로 Programmers에서 다양한 문제를 해결하면서 구현 능력을 키웠습니다. Java Servlet과 JSP을 이용해 순수 자바로 게시판 및 To-Do 리스트 등의 간단한 페이지나 Web socket등을 이용해 간단한 숫자 야구 게임을 개발해본 경험이 있습니다.

Spring Framework / Spring Boot

Spring Framework의 특성을 이해하고 있으며 복잡한 구성 및 설정을 단축화 하기 위해 Spring Boot를 통해 프로젝트를 진행했고 Restful하게 Rest API서버를 구축해 배포하였습니다. Spring Doc를 기반으로 Swagger를 사용해 API명세를 작성해본 경험이 있고 이를 통해 Api 검토 및 협업을 하는 과정에서 필요한 이유를 알 수 있었습니다.

Data

RDBMS (Oracle Database, MySQL)

RDBMS 중 MySQL을 이용하여 팀 프로젝트를 진행 해 봤고 Key 와 Index 의 장단점을 알고 있습니다. ERD설계부터 SQL query를 통해 데이터를 잘 활용할 수 있습니다.

Redis

메모리에 저장되는 특성과 Redis의 비정형 데이터 구조 중 하나인 sorted-set을 이용하여 단위 시간의 인기 검색어 top 10을 휘발성으로 저장하고 조회하는 기능을 구현해 본 경험이 있으며, Redis의 특성이 타 DB와 어떻게 다른지 이해하고 있습니다.

Deployment

Linux / AWS

저전력 미니pc에 Cent-OS를 이용하여 집에서 간단한 포트폴리오 서버를 운영해보고 있고 Linux의 여러 환경 설정을 통해서 서버의 보안 및 운용 환경을 조절해보고 있습니다. 팀 프로젝트를 진행하면서 AWS EC2 서버를 이용하여 배포해 본 경험이 있습니다.

Docker를 통해 AWS EC2 서버에 배포를 해봤습니다. Docker를 통해서 배포 환경을 개발 환경과 계속 맞춰 줄 필요 없이 배포가 가능함을 알게 되었고, Docker-Compose를 통해서 여러 이미지로 컨테이너를 실행하여 서버 구축의 편의성을 경험했습니다.

Front End

HTML5, CSS3, Javascript

Bootstrap

Vue.js

Vue.js의 랜더링 메커니즘인 V DOM과 생명주기 혹은에 대한 개념을 이해하고 있고 이를 적용하여 csr 형식의 웹 프로젝트를 제작 할 수 있습니다. vue router, vue cookies 같은 라이브러리를 사용하여 현재의 상황에 따른 페이지 및 정보를 제공할 수 있고, 커스텀 컴포넌트, 커스텀 이벤트를 이용하여 코드의 중복을 줄이고 대댓글 같은 기능은 자기 참조의 형식을 이용해 구현해 봤습니다.

Tools

Git, Git Hub, Source Tree, Zira, Postman

About Me

새로움에 두려움 없이 도전하는 사람

무언가 새로 시작한다는 것은 이전까지 익숙하게 해왔던 것을 한 편에 묻어두고 미숙한 부분을 통해 나 스스로를 성장 시킨다는 의미라고 생각합니다.

미지에 대한 두려움, 미숙한 실력부터 다시 시작해야 한다는 것은 꽤 견디기 힘든 일입니다.

하지만 두려움과 부담감을 넘는 다른 감정을 가진다면 새롭게 시작하는 것은 꽤 기분 좋은 일 수 있습니다.

저는 제 적응력과 습득력이 좋다고 생각하고 있고 이 장점을 통해서 분명 새롭게 시작한 미숙한 일이라도 금방 습득하고 숙련된 업무 처리를 가능케 할 수 있다고 믿습니다.

이러한 믿음으로 저는 새롭게 시작하는 것에 대해서 두려움을 가지는 것이 아닌 자신의 성장을 기대하고 도전할 것입니다.

이해하기 쉬운 코드를 위해 수정하는 개발자

좋은 글이 되기 위해서는 두 가지 조건이 필요하다고 생각합니다. 내용이 좋아야 하고 다른 하나는 읽기 편해야 합니다. 즉 이해하기 쉬워야 된다는 것입니다.

코드 또한 컴퓨터도 같이 이해하는 하나의 언어로 작성된 글이기에 이해하기 쉽게 작성해야 하는 것이 좋은 코드를 작성하는 것이라고 생각합니다.

개인 프로젝트를 진행하더라도 나중에 다시 내가 짠 코드를 봤을 때 바로 이해가 가능하게 코드를 작성 할 수 있도록 노력하고 있습니다.

특히 팀 프로젝트를 진행하면서 팀원들이 제가 맡은 부분에 대해서 충분히 이해할 수 있도록 수정하고 또 수정하며 가독성이 높은 코드를 작성하고 싶습니다.

마무리 짓지 않으면 잠들지 못하는 개발자

어떤 일이던 시작하게 되면 특이점이 오기 전까지는 앞만 보고 달리는 경향이 있습니다.

간단한 기능의 경우에는 특이점이라고 할 것도 없이 해당 기능을 전부 개발하고 복잡한 기능이라면 해당 기능을 분류 하여 각각의 분류에 대한 개발을 달성하는 것을 목표로 개발합니다.

가끔은 개발하다가 아이디어가 잘 떠오르지 않거나 떠오른 아이디어를 구현하는 데에 막혀 오랜 시간이 소요될 때가 있습니다. 다음 날이 중요하지 않다면 해당 부분을 마무리할 때까지 개발하고 이를 달성했을 때의 성취감을 얻곤 합니다.

하지만 어쩔 수 없이 잠을 청해야 하는 경우가 있습니다. 이럴 때는 정말 잠들지 못합니다. 잠에 들기 마지막까지 어떤 식으로 개발해야 성공적으로 완수 할 수 있을지 시뮬레이션을 계속 돌리면서 잠을 설치기 일쑤입니다.

개발할 범위가 정해지게 되면 이를 달성하기 위한 굳은 의지를 가지고 개발에 임하고 있습니다.

Project

여행한 지도 2024.02.22 ~ 2024.03.31

<https://github.com/s12171934/trip-one-zido>

역할

- Team Leader, PM
- 프로젝트 기획 및 일정 계획, Git Branch 관리
- DB ERD 설계, DB 쿼리 설계
- 화면 설계, API 설계, 프로젝트 구조 설계
- Rest API 서버 Controller 및 Swagger을 통한 API 명세 제작
- Vue.js를 통한 View단 개발
- SNS 연동 로그인 등 외부 API와 연결이 필요한 부분 개발
- 전반적인 오류 사항 파악 및 수정

사용 기술

- Java 17, Spring Boot, MySQL, Redis, Vue.js, Bootstrap, Git, Docker, AWS

프로젝트를 진행하며 경험한 부분

- Team Leader로써 Git Hub와 관련된 부분을 관리하면서 겪은 고찰
 - 각 인원이 각각의 자신의 브랜치를 가지고 코드를 작성, 개발하다가 서로의 작업물이 필요한 경우에는 오직 pull request를 통해서 merge가 이루어질 수 있도록 합의했습니다.
 - 간혹, main에 잘못 PR하는 경우가 있어 이를 방지하고자 팀원들의 코드 리뷰가 진행되고 난 뒤에만 request를 승인 할 수 있도록 규칙을 추가 했습니다.
 - 거의 팀원들이 오프라인 상에서 함께 모여 개발을 진행하다 보니 전체적인 업무 분담과 코드 리뷰의 경우는 오프라인 상에서 진행되어 이슈 발행 및 PR의 Comment가 부족했었습니다.
 - 개발 도중 Squash Merge에 대해서 알게 되었고, 이전에 잦은 commit으로 인해 찾기힘들었던 중단점을 더 잘 찾고, 브랜치가 비교적 깔끔하게 관리될 수 있게 되도록 Squash Merge를 한 이후에 PR을 할 수 있도록 진행했습니다.
- 프로젝트 패키지 구조의 변화에 따라 개발 및 검토의 방식에 변화가 있었습니다.
 - API 서버의 패키지 구조를 controller, service, mapper로 나누어 개발하였으나 팀원들 간의 합의를 통해 각 기능 별로 구조를 나누는 방식으로 변화를 주었습니다.
 - 기존에는 계층 별 검토를 진행했고 여러 기능에 대한 수정이 동시에 이루어지며 왔다 갔다 하는 시간이 더 많았습니다.
 - 변경 이후 각 기능 별 수정이 필요한 부분이 더 빠르게 파악되었고 더 효율적으로 개발이 가능했습니다.
- 미리 정해둔 DTO를 통해 일정한 양식의 틀을 따라가며 개발할 수 있었습니다.
 - 전체적으로 기능을 구현하기 전 API명세, query를 통해 얻을 데이터를 미리 정의해 필요한 DTO명단을 미리 작성해뒀고 해당 DTO에 맞춰 각 계층 별 통신을 이루었습니다.
 - 협업을 하는 과정에서 어떤 형식으로 데이터를 가져와야 하는 지에 대한 궁금증을 많이 줄일 수 있었고 자세한 컨벤션이 부여되지 않았음에서 비교적 일정한 결과 값을 얻을 수 있었습니다.
- 중복된 부분을 줄여 추후에 유지 보수에서 이점을 기대할 수 있었습니다.
 - 대표적으로 Vue.js에서 알림 모달이 있습니다. 형식이 같은 모달들을 컴포넌트로 틀만 제작하고 각 모달의 데이터를 외부에 저장했습니다.
 - 추후에 모달을 추가할 때에는 각 모달의 데이터만 리스트에 추가하면 바로 사용할 수 있었습니다.