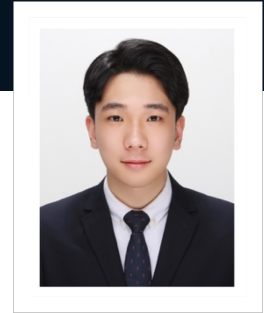


# 이혁준 SOFTWARE ENGINEER



**BIRTH** 1996. 05. 13  
**ADDRESS** 부산시 부산진구 성지로 10-1  
**PHONE** 010-4751-5344  
**EMAIL** [biper94@gmail.com](mailto:biper94@gmail.com)  
**Portfolio** <https://2hyjun.github.io/>

## 업무적 강점

- React-Native를 활용한 Front-End 개발 경험
- ExpressJS, AWS 인프라를 활용한 Back-End 개발 경험
- 먼저 스스로 해결하려는 자세

## 경력 요약

2017. 10 – 2018. 10 (주)샤프고트 연구개발팀 사원 IT 개발 총괄

## 학력 사항

2015. 03 – 현 재 부산대학교 정보컴퓨터공학과 재학  
2012. 03 – 2015. 02 개성고등학교 졸업

## SKILL SET

Language Javascript(Typescript), Nodejs  
OS MacOS  
Infra AWS, Firebase  
Server ExpressJS, React Native  
DataBase MySQL  
Development Tool VSCode, WebStorm, Git

## 상세 경력

2017. 10 – 2018.10 (주)샤프고트 IT 개발 총괄

## 주요 업무

- IoT 하드웨어 제어, 실시간 모니터링 가능 모바일 앱 개발
- IoT 하드웨어 부착 MQTT 서버 개발
- Toy Project 시제품 펌웨어, 모바일 앱 개발

---

## 자기소개서

### [주먹 쥐고 일어서다]

무슨 문제가 생기든, 어떻게든 해결하고 이겨내는 것이 저의 가장 큰 장점입니다. 평범하지 못한 가정환경에서 힘들게 자랐지만, 사회와 인생을 닦으며 도태되지 않았고 항상 상황을 극복하려 노력했습니다. 고등학교 시절 국가지원에 문제가 생겨 큰 빛이 생겼을 때, 해결책을 찾아 빛을 줄이고 아르바이트를 하며 해결한 경험이 있습니다. 이런 경험은 역경을 느끼고 슬퍼하는 것이 답이 아니라, 문제를 해결해야 한다는 점을 알려주었습니다. 또한 이전의 경험들을 밑거름 삼아 아무리 힘든 상황도 이겨낼 수 있다는 자신감을 가지게 되었습니다. 이런 점을 이어나가, 개발 이슈가 생기면 Reference를 정독하고, StackOverflow 등을 통해서 스스로 방법을 찾고 해결하는 능력을 기르게 되었습니다.

### [취미에서 진로로]

이런 성장과정 중 저의 취미는 컴퓨터 게임이었지만, 컴퓨터가 좋지 않아 게임하기 힘들었던 기억이 납니다. 그래서 사양 최적화, 컴퓨터 업그레이드를 통해서 해결하려 했습니다. 그 과정에서, 수많은 프로그램을 사용했고 많은 시스템 설정을 건드렸습니다. 자연스럽게 소프트웨어와 친해졌던 것 같습니다. 게임을 만들어보고 싶어, 어도비 플래시를 배워서 코딩하며 게임을 만들었고 안드로이드를 더욱 빠르게 사용하고 싶어 루팅하고 커스텀 롬, 커널을 설치하며 스마트폰을 커스터마이징하는 것을 즐겼습니다. 소프트웨어에 대한 관심과 열정으로 자연스레 컴퓨터공학과에 진학하였고, 개발자로 진로를 선택하게 되었습니다.

### [남다른 끈기]

대학교에서 파일구조의 Dynamic Hash 프로젝트를 할 때, 50만개의 샘플 데이터를 읽고 indexing 하는 데 2시간이 걸리는 이슈를 해결하지 못하는 상황이 있었습니다. 위 이슈를 해결하기 위해, 50만개의 샘플데이터 Row 하나하나 걸리는 시간을 측정하고, 특이하게 오래걸리는 Row가 없는지 확인하였습니다. 또한 사용한 알고리즘들을 분석하여 Time Complexity를 최소화 할 수 있는 방법을 찾으려 노력했습니다. 3일동안 밤을 새며 노력하여 결국 새로운 File Block을 생성할 때 불필요한 작업이 존재하는 것을 깨닫고 해결하였습니다. 끈기 있게 시행착오를 피하지 않았고, 그 시행착오를 통해 경험이 쌓여가며 저를 더욱 단단하게 만들었습니다. 끈기를 통해서, 무엇이든 빠르게 배울 수 있었습니다. 모바일 앱을 통해서 창업을 하는 동아리에 가입해서, 그 곳에서 React-Native 모바일 앱 제작 프레임워크를 익혔습니다. React-Native를 처음 접한지 5개월 만에 완성도 있는 모바일 앱을 개발하였고, 백엔드에 대한 스터디도 병행하여 백엔드도 구축하였습니다. 또한, 개발 부장을 역임하여 동아리 내 React-Native 기반 모바일 어플리케이션, Express 및 MySQL 기반 API 서버 개발 스터디를 진행하였습니다.

### [뛰어난 업무수행능력과 문제를 스스로 해결하는 인재]

대학교에서 쌓은 경험을 통하여 스타트업에 입사하여 많은 프로젝트의 개발을 총괄하였습니다. 실제 프로덕션 레벨의 프론트엔드, 백엔드 개발 뿐만 아니라, 시제품 회로를 제작하고 펌웨어를 개발하였습니다. 스타트업 회사에서는 개발에 관해서 회사 내에서 자문을 구하거나 도움을 요청할 사람이 없었습니다. 문제가 생기면 스스로 해결하려 노력하였고 해결할때까지 멈추지 않았습니다. 그 결과, 혼자서 프론트엔드 모바일 앱과 백엔드 API서버 등을 개발할 수 있었습니다.

저는 큰 꿈을 가졌기에 현실에 안주 하지 않고 도태되지 않으며 항상 발전을 추구하는 인재입니다. 저의 지난 경험과 진취적인 마인드는 충분히 능력으로 인정받을 수 있을 것이라 자부합니다

## 프로젝트 상세 수행 이력

2018. 03 – 2018.10

(주)샤프고트

IT 개발 총괄

- 프로젝트 명: IoT 프로덕션 레벨 모바일 앱, 서버 개발

### 수행 역할

IoT 제조업 스타트업에서 실제 양산 프로젝트에 참여하여 프론트엔드, 백엔드 개발을 총괄하였습니다. 디바이스를 제어하고 모니터링 가능한 모바일 앱, 사용자 및 디바이스 정보를 관리할 DB 서버와 API서버, IoT 디바이스를 제어하고 데이터를 중개하는 서버를 개발하였습니다.

#### [Back-end: MQTT Server]

본 프로젝트를 시작하며 가장 고민하였던 것은, “디바이스를 제어하고, 모니터링 할 시스템 구조를 어떻게 설계하는 것이 좋을까?”였습니다. “MQTT 프로토콜”을 이용하는 것이 가장 최적의 방식이라고 판단하였습니다.

초기에는, NodeJS의 “mosca” 라이브러리를 통하여 직접 MQTT 서버를 구현하고 운영하려 하였지만 클러스터링에 대한 제약사항 등을 고려하여 AWS 사의 AWS IoT를 이용하는 것이 바람직하다고 생각되었습니다. AWS IoT를 이용하며 AWS Cognito, Lambda 등을 활용하며 AWS 인프라에 대한 이해도를 증진시킬 수 있었습니다.

#### [Back-end: API Server]

API 서버를 개발하며 가장 큰 고민은 “어떻게 서버가 수십만명이 넘는 사용자를 수용할수 있게 하지?”였습니다. 그 고민을 해결하는데 AWS를 적극 활용하였습니다. AWS 인프라 내에서 Elastic Load Balancer, Auto Scaling을 적극 활용하였고, NodeJS의 PM2라이브러리를 통해서 클러스터링을 관리하였습니다. Back-end 인프라를 설계하고 관리한 점은 Back-end 인프라 더 넓게 바라보고, 큰 그림을 그릴 수 있도록 저에게 큰 성장의 밑거름이 될 수 있었습니다.

#### [Front-end]

IoT 디바이스를 제어하고, 모니터링이 가능한 앱을 개발했습니다. AWS IoT와 연결된MQTT Client를 통해 메시지 수신 시 메시지를 파싱한 후 State에 반영하여 현재 상태를 출력하는 기능을 개발하였고 기기 제어를 위하여 MQTT Client로 메시지를 송신하는 기능을 개발하였습니다..

주 기능 외에도, 로그인/회원가입 등 회원정보 관리 페이지, 디바이스 등록/수정 등 디바이스 관리 페이지를 개발하였습니다. 그 중 디바이스 등록 파트는 스마트폰의 Wi-Fi 를 제어하고 HTTP Request를 통하여 디바이스와 통신하는 기능이었습니다. Android의 자동으로 Wi-Fi에 연결하는 메소드를 활용하여 직접 NativeModule를 구현하여 사용하였습니다.

Redux State 관리를 사용하여Smart Component와 Dumb Component를 구분하며 개발하여 Data flow를 최적화 하도록 노력했고 ES2017의 async와 await을 최대한 활용하여 비동기 프로세스를 제어하였습니다.

가장 큰 고충은 여러 스마트폰에 맞게 UI를 제작하는 것었습니다. 스마트폰의 비율, 화면 크기등을 생각하여 Styling을 하여도 UI가 깨지거나 찢리는 현상이 있는 스마트폰이 있었습니다. 화면 크기 및 비율에 따라 폰트 크기, 여백 크기 등을 분기하여 처리하면서 해결하였습니다.

그간 연습했던 제 기술들을 모두 발휘한 프로젝트였습니다. 프로젝트 중에도 부족한 점을 인지하며 많은 스터디를 진행하였고 그 과정에서 Typescript, GraphQL, Mobx 등 사용해보지 못한 기술들을 접하며 또 다른 스터디에 대한 계획을 세우기도 한 프로젝트였습니다.

- 프로젝트 명: Toy Project 시제품 펌웨어, 모바일 앱 개발

## 수행 역할

IoT 제조업 스타트업에 취업하여 가장 먼저 맡은 역할이 시제품을 만드는 것이었습니다. Arduino, ESP8266, RaspBerry PI 등을 활용하여 회로를 제작하였고, 해당 MCU에 맞는 펌웨어를 개발하였습니다. 그리고 디바이스를 제어하는 인터페이스 용도와 디바이스를 모니터링 할 수 있는 모바일 앱 또한 개발하였습니다.

### [임베디드: 회로 제작, 펌웨어 제작]

NodeMCU, Arduino, RaspBerry PI를 활용해서 회로를 구성하고, 그에 맞는 펌웨어를 설계하였습니다. IoT 기술을 기반하는 시제품을 제작하는 것이 주 목표였기 때문에, Wi-Fi 혹은 Bluetooth 혹은 Zigbee를 활용하여 외부에서 기기를 제어하고, 외부로 디바이스 상태를 모니터링 하게 하는 제품이었습니다. 주로 디바이스를 제어하는 방식은 Wi-Fi 라이브러리를 통하여 HTTP 서버를 통해서 외부에서 제어할 수 있도록 하였습니다. 블루투스의 경우는 Serial 통신을 활용하여 외부와 디바이스 간의 통신을 관리하였습니다.

### [모바일 앱 제작]

React-native를 활용하여 IoT 디바이스를 제어하는 모바일 어플리케이션을 제작하였습니다. React-Native은 UI 개발 생산성이 뛰어난 프레임워크이기 때문에 UI 개발은 무난히 진행 할 수 있었지만, Wi-Fi 자동연결, Bluetooth Serial 통신 등은 기본 라이브러리로 제공되지 않는 기능이기 때문에 구현하기가 순탄치 않았습니다. 결국 두가지 모두 Native하게 구현하여 이용하거나, 오픈소스 라이브러리를 그대로 이용 혹은 수정하여 이용해야 했습니다.

Wi-Fi 제어(Wi-Fi 자동 연결, AP 목록 스캔 등)은 Android에서는 WifiManager 등 기본으로 제공되는 Android 클래스가 있었고 오픈소스 라이브러리가 존재하여 구현할 수 있었지만, 라이브러리의 완성도가 높지 않아 불안정 하였습니다. 그래서 해당 라이브러리를 Fork하고, 수정하고 재배포하는 과정을 거쳐 구현할 수 있었습니다.

Bluetooth Serial 통신 또한 Android에서 기본으로 제공되는 클래스를 이용하여 구현 가능하였습니다. 하지만 iOS에서는 Bluetooth 4.0 (BLE)만 지원되기 때문에 Class Bluetooth(Serial 통신)은 구현 할 수 없었습니다.

그동안 학교에서 접해보지 못한 회로를 다루면서 가장 크게 느낀점은 “배워서 하면 되는 구나”였습니다. 대학교에서 전기회로 등 이론수업을 들을 때는 하드웨어를 다루는 것에 대해 상당히 부담감을 느꼈지만, 책임감을 느낌에 따라 느꼈던 열정을 통하여 스스로 공부하고 배우다보니 어느새 능숙하게 다룰 수 있었습니다.

그리고 Toy Project를 진행하며 단기간에 많은 모바일 앱을 만들면서, React-Native에 대한 숙련도를 한층 끌어올릴수 있었습니다. 실제 양산 제품용 모바일앱을 만들기 위한 충분한 성장의 과정이 되었다고 생각합니다.

- 프로젝트 명: Pretzel

## 수행 역할

모바일 앱을 제작하여 창업을 하는 대학동아리에서 팀을 이루어 프로젝트를 진행하였습니다.

“Pretzel”이라는 대학교 내 스쿠터족을 활용한 사용자간 배달 매칭 서비스를 개발하였습니다. 저는 해당 앱의 Back-end개발을 총괄하였고, Front-end 개발에도 상당 부분을 가담하였습니다.

### [Back-end]

Back-end는 DB서버로 접근을 관리할 웹서버(API 서버), 사용자간 채팅을 관리할 채팅서버, 사용자 및 서비스 정보를 저장할 DB서버를 개발해야 했습니다. 모두 오픈소스 프로젝트를 활용하여 개발하였습니다.

### [Back-end: Web Application]

API 서버는 ExpressJS 웹 어플리케이션 프레임워크를 활용하였습니다. 사용자의 인증토큰 관리, Database data Query 등 주요 기능을 구현하였습니다.

그 중, DB서버로의 Query 수를 줄이기 위하여 SQL문을 최적화하는 데 노력하였고, 그 과정에서 Relational Database의 “JOIN”이라는 메소드의 장점과 Database 스키마 설계에 대한 중요성을 깨닫게 되었습니다.

### [Back-end: Chat Server]

채팅서버는 실시간 양방향 커뮤니케이션 라이브러리인 Socket.io를 활용하여 개발하였습니다. 사용자간의 채팅 메시지를 관리하여야 했고, offline 유저 메시지

사용자의 접속 시점 관리, 접속 유지 시간 관리, 메시지 관리, offline 유저 메시지 전달을 관리하는데 노력하였습니다.

### [Front-end: React-Native]

React-Native를 활용하여 크로스플랫폼 모바일 어플리케이션을 개발하였습니다. 사용자의 계정정보 관리(로그인 및 회원가입), 게시물 CRUD(작성, 조회, 수정, 삭제), 사용자간 채팅 등 주요기능과 UI구현을 담당하였습니다.

React-Native를 처음 접한지 2달만에 실제 프로젝트에 참여하게 되었습니다. React-Native를 사용하면서 기존 Native플랫폼 개발과 차원이 다른 UI 개발 생산성, 그리고 Android/ iOS플랫폼 개발을 동시에 진행할 수 있는 점에 이내 매료되었고 이로인하여 빠르게 개발을 진행할 수 있었습니다.

또한 npmjs.org, github 커뮤니티에 업로드된 여러 오픈소스 라이브러리를 활용하여 개발 하였고 그 과정 중 타 개발자의 코드를 보고 배우고, 수정하고, 기여하며 “오픈소스”에 대한 장점을 깨닫게 되었습니다.

그간 동아리에서 스터디/세미나를 통하여 배웠던 Front-end, Back-end기술들을 처음으로 실무에 활용했던 프로젝트였습니다. 주요 프레임워크/라이브러리 사용법, 협업 방법 등의 기초 개념을 익히기에 충분했던 경험이었습니다. 또한 “맨땅에 헤딩하는 법”을 배웠던 좋은 경험이었다고 생각합니다. Document, Reference를 읽는 법, 질문하는 법 등 굳이 남이 가르쳐주지 않아도 혼자서 해결할 수 있는 방법을 터득할 수 있었습니다. 이후에 스타트업에 입사하여 실무를 경험하고 헤쳐나가는 데 중요한 기반이 되었습니다.

지원서 상의 모든 기재 사항은 사실과 다름이 없음을 확인합니다.

2018 년 10 월 24 일

이 혁 준