

JUNIOR DEVELOPER

PORTPOLIO

> About Me



김성훈

LICENSE

- SQLD 2024.04
- 정보처리기사 2023.11
- 토익스피킹 IM1 2023.11

EDUCATION

- 조선대학교 전자공학과 졸업 2024.02
- 산대특 IoT 개발자 과정 수료 2024.08

SKILLS

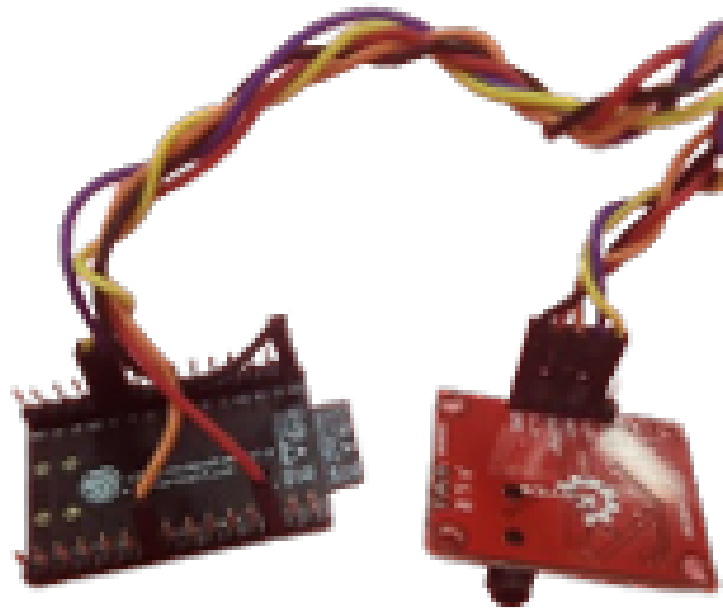
- Python, C++, Dart
- Flask (Python), Node.js, RESTful API
- AWS, SQL
- RESTful API, MQTT, HTTP/HTTPS
- Linux, Git

> Project

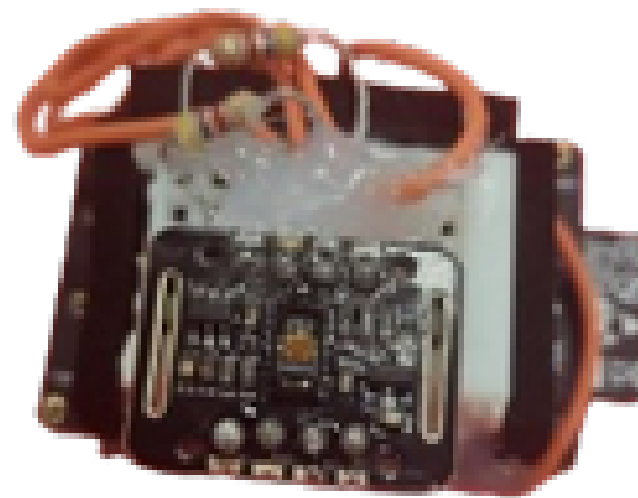
딥러닝 기반 생체신호 분석 및 부정맥 검출 앱 서비스

웨어러블 기기를 통해 실시간 심전도 신호를 수집하고,
심장 지표를 모니터링하여 부정맥을 검출하는 서비스

ECG 회로 구성



PPG 회로 구성

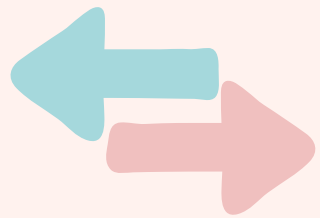


실제 제품 사진



> Build & Resolve

IoT 디바이스를 활용한 실시간 데이터 수집 및 송신을 위한 통신 구현



I2C 및 SPI



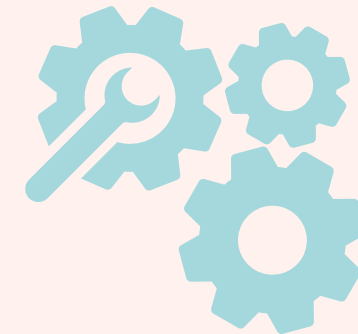
Wi-Fi,
AWS S3



BLE,
Flutter



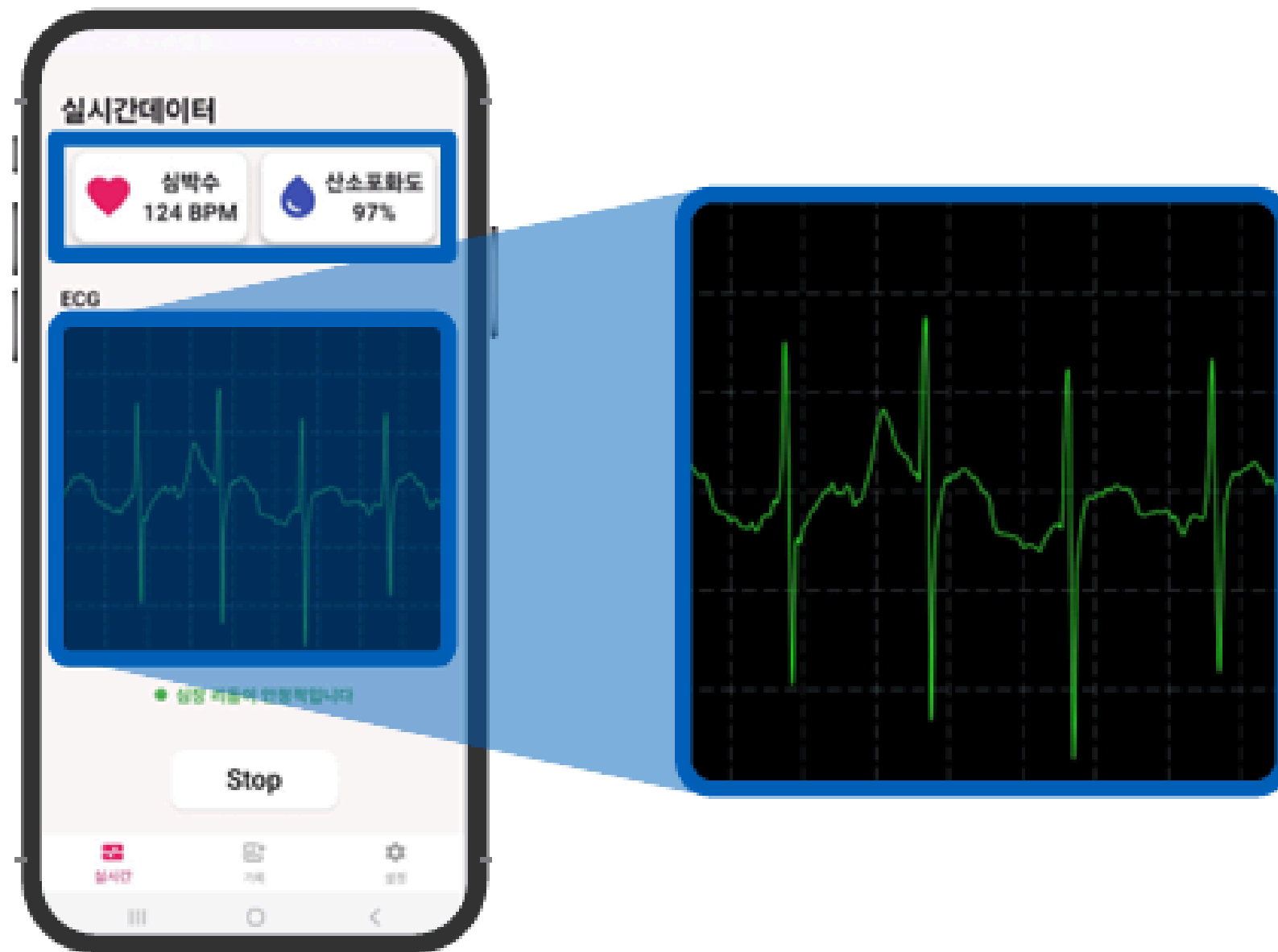
신호 데이터
전처리



트러블 슈팅

> ESP32, Flutter, BLE

Node.js와 Restful API, Flutter의 WebView를 활용하여 구글 및 소셜 로그인을 구현



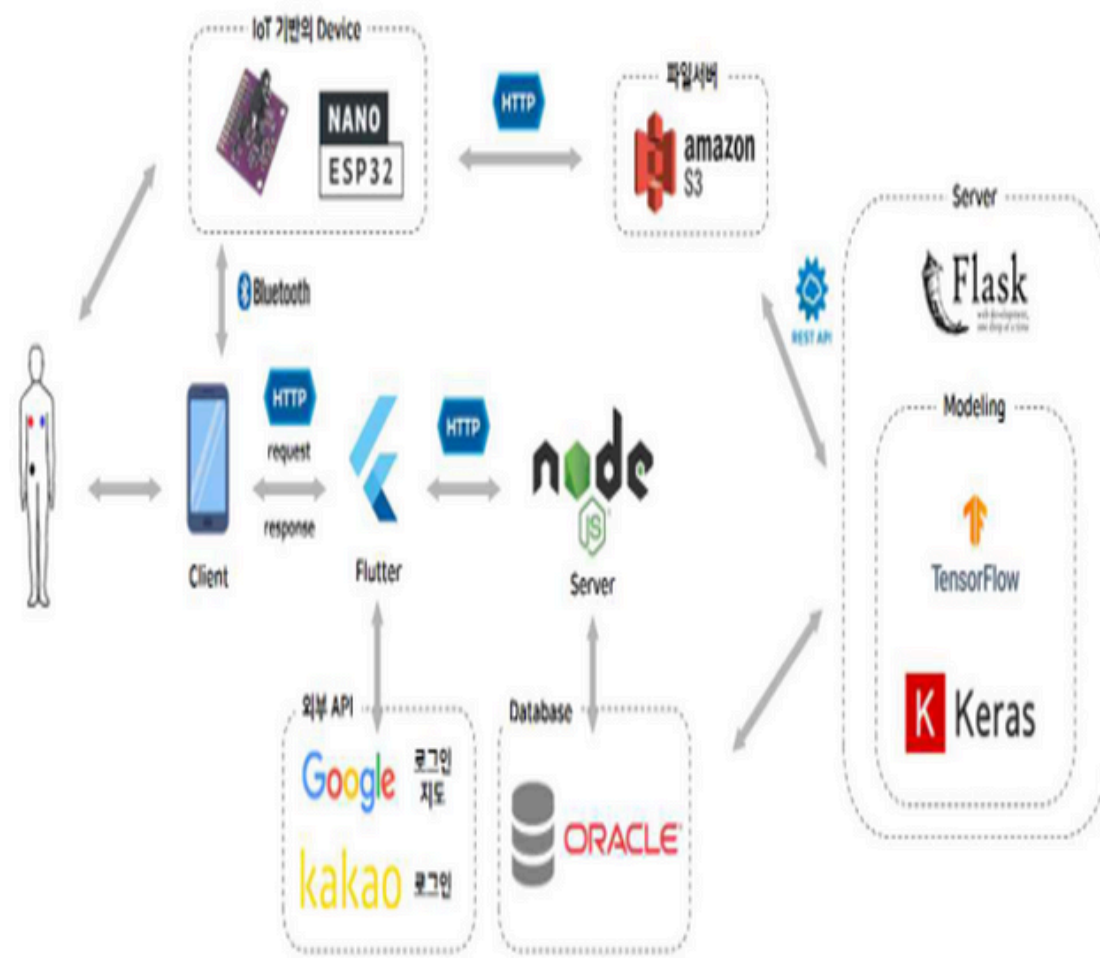
Blue_plus 라이브러리를 사용해
생체신호 데이터 수신



fl_chart 라이브러리를 사용해
움직이는 심전도 그래프 구현

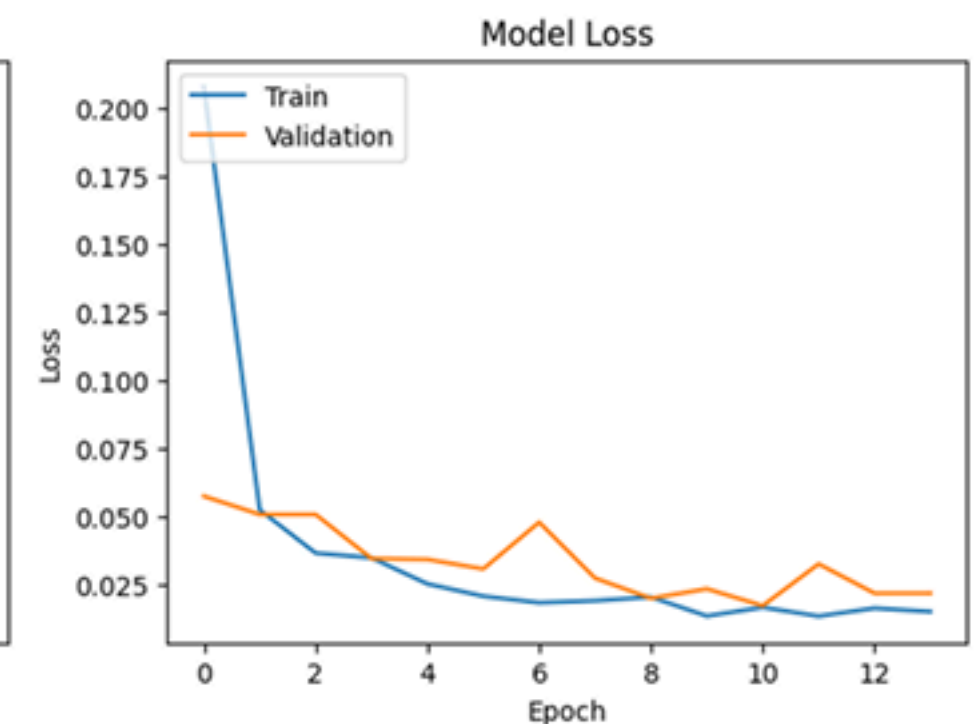
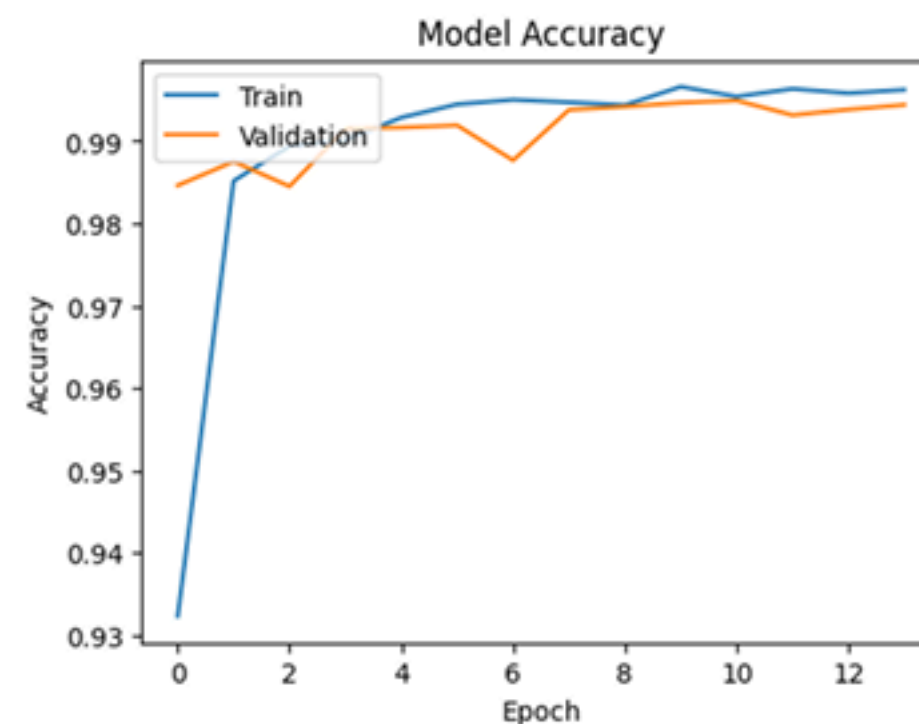
> ESP32, AWS, Wi-Fi

Wi-Fi 통신을 통해 ESP32와 API Gateway, AWS S3를 활용한 데이터 전송 및 저장 구현하여 부정맥 검출 모델링을 통해 실시간 ECG 데이터 예측 및 결과 도출



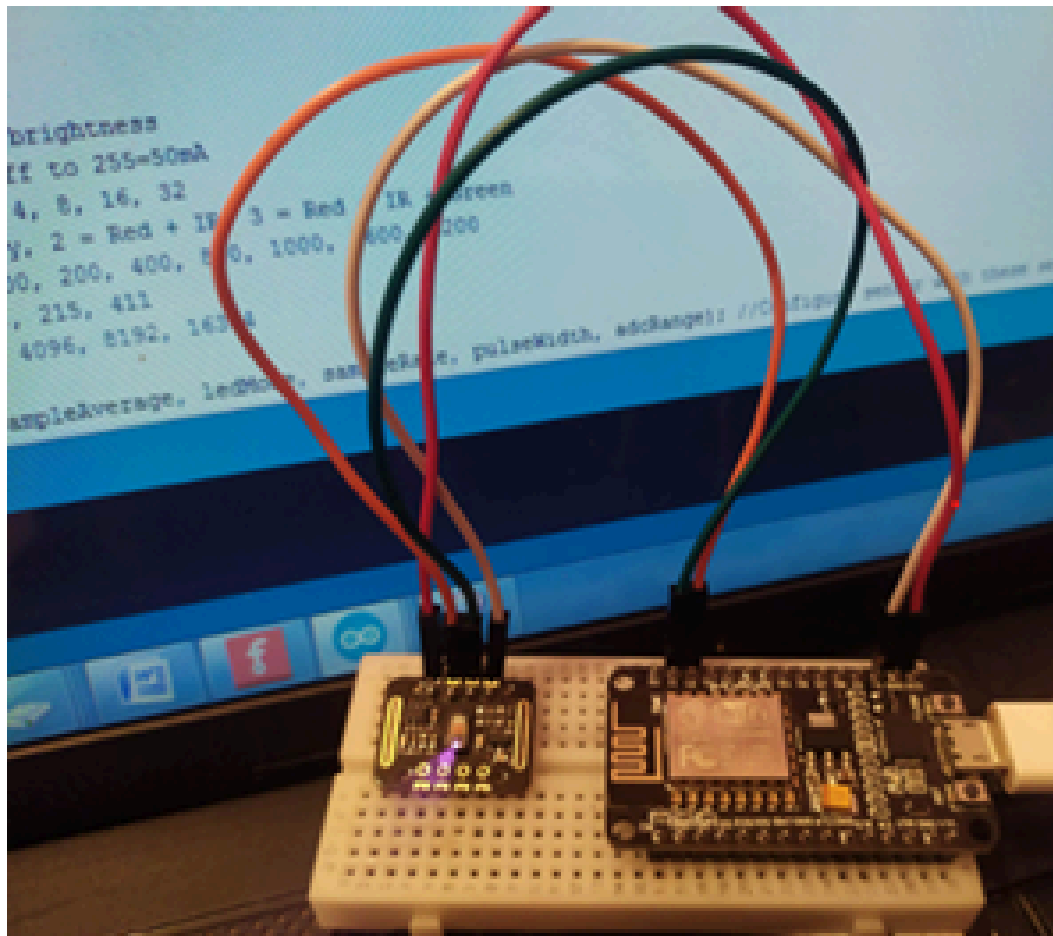
accuracy: 0.9952
val_accuracy: 0.9946
loss: 0.0189
val_loss: 0.0282

Hyperparameter	Value
Loss function	SparseCategoricalCrossentropy
Optimizer	Adam
Batch size	32
Learning rate	0.001
Epoch	10



> 트러블 슈팅

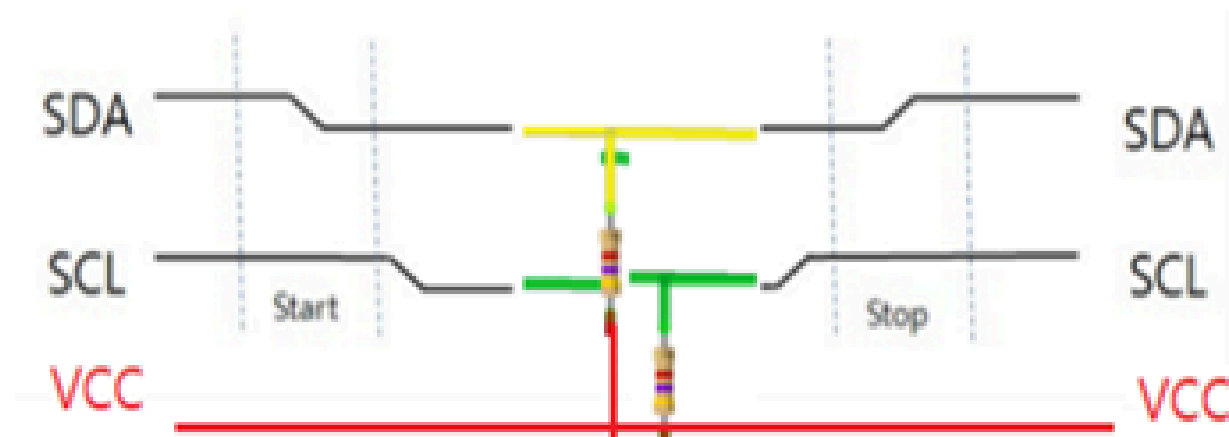
I2C 통신 오류



내부 풀업 저항이 충분하지 않아
I2C 통신 오류 발생



SDA와 SCL핀에 4.9k Ω 저항을
VCC에 연결하여 외부 풀업 저항을 구성

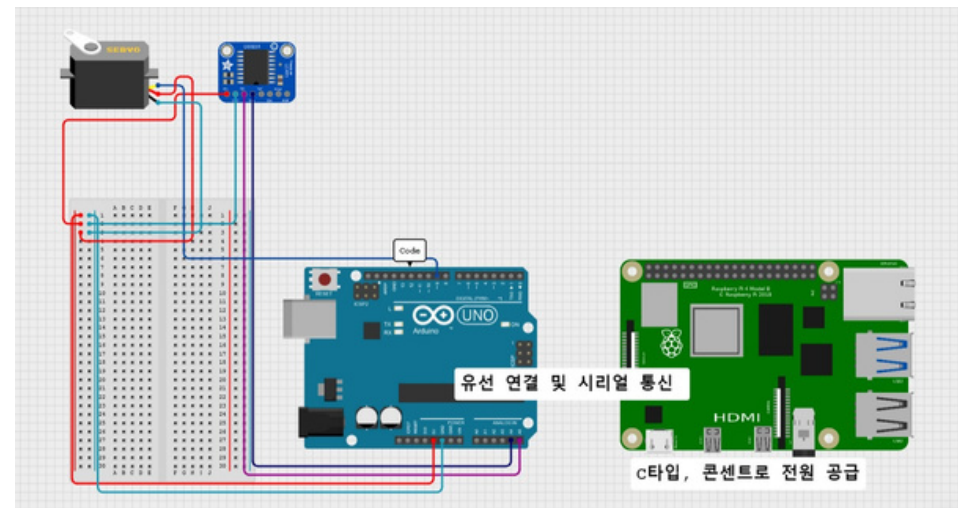


> Project

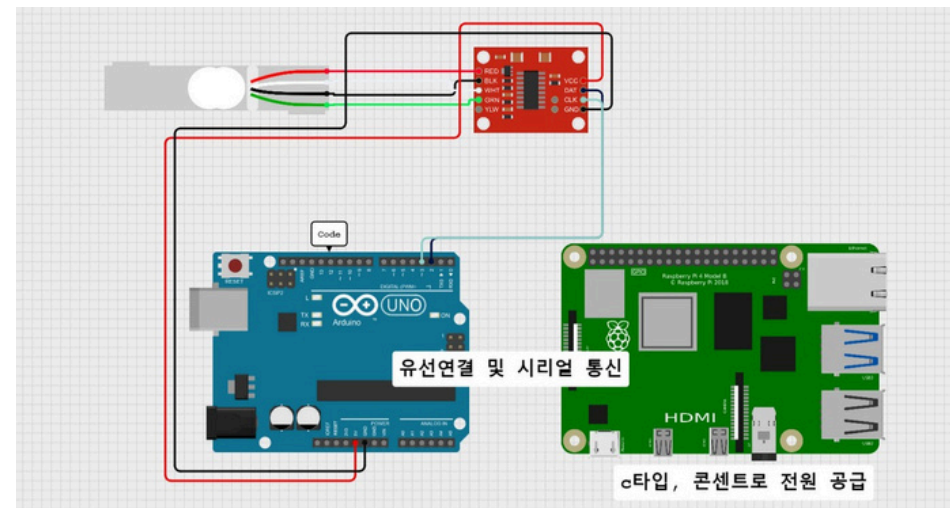
스마트홈 기술을 활용한 반려동물 케어 웹 서비스

웹페이지를 통해 반려동물의 사료 제공을 관리하고,
체중 데이터를 수집하여 시각화하며, 커뮤니티 기능을
통해 사용자 간의 정보 공유를 지원

자동배식기 회로도



체중계 회로도



실시간 연동 페이지

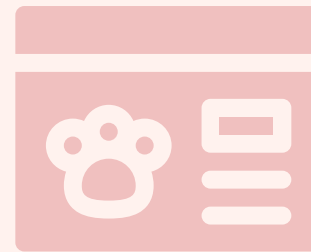


> Build & Resolve

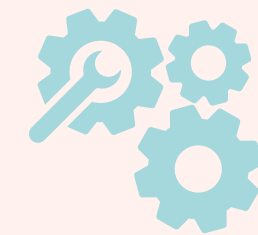
IoT 기술을 웹페이지와 연동하여 실시간 데이터 제공 및 자동화된 반려동물 돌봄 서비스 제공



실시간 데이터
웹 연동



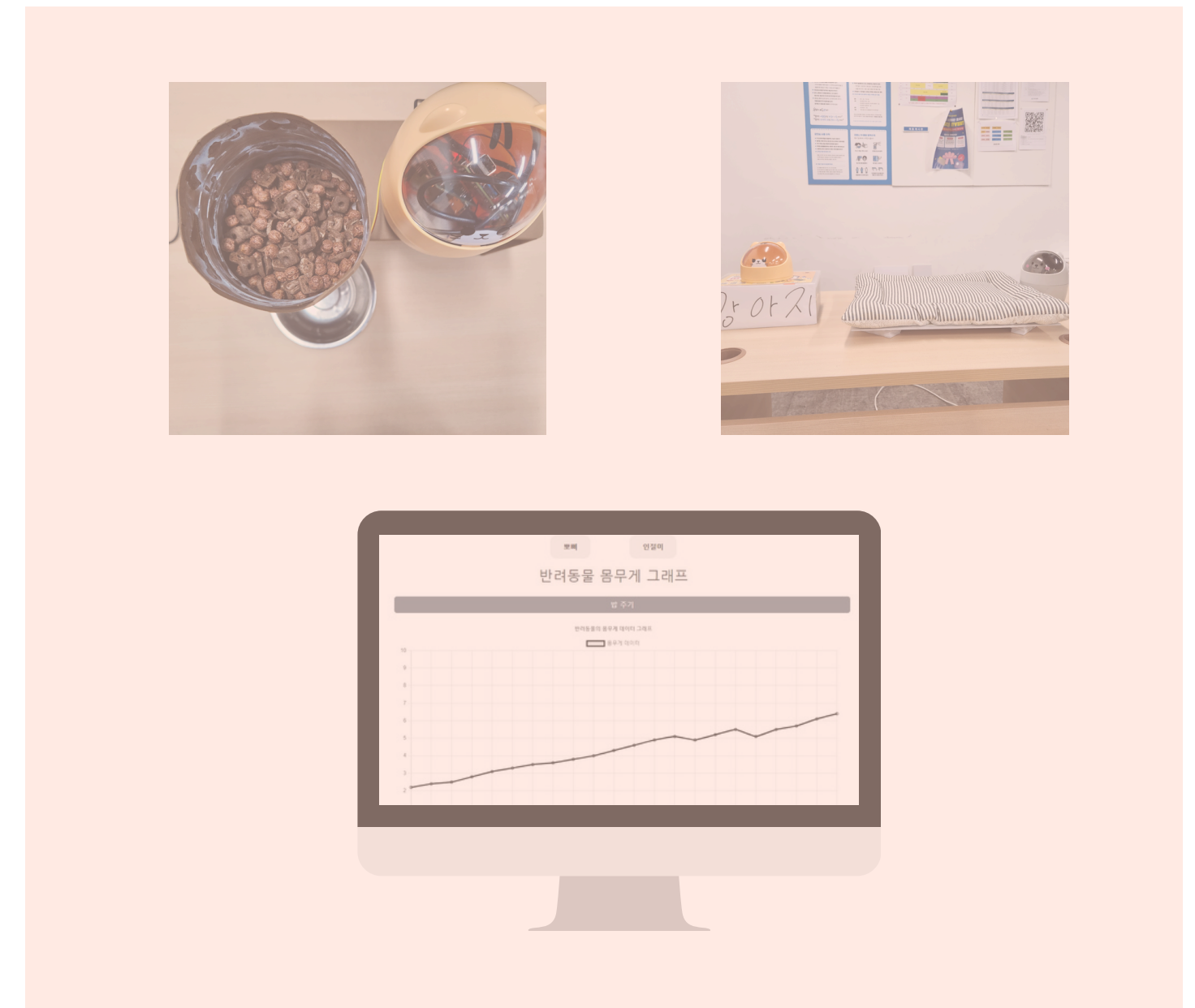
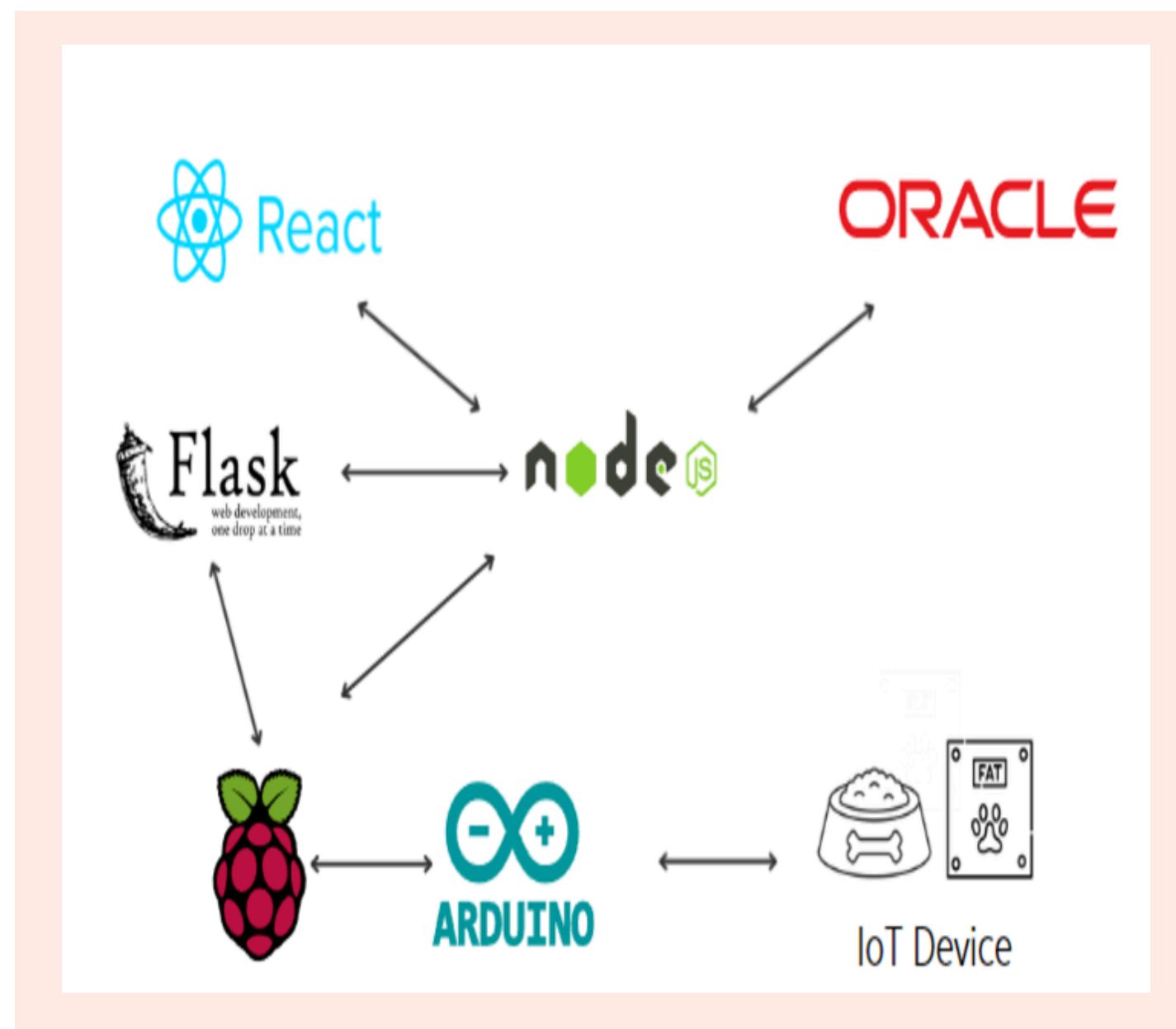
펫 정보 관리



트러블 슈팅

> 실시간 데이터 웹 연동

IoT 기술을 웹페이지와 연동하여 실시간 데이터 제공 및 자동화된 반려동물 돌봄 서비스 제공



> 펫 정보 관리

반려동물의 체중 데이터를 수집하여 시각화하고, 여러 반려동물의 데이터를 효율적으로 보여주기 위해 Pagination 기능을 구현

```
const Board = () => {
  const dogsession = JSON.parse(sessionStorage.getItem("user"));
  // 게시판 관련 관련 변수
  const MAX_LENGTH = 30;
  const [board, setBorad] = useState({
    title : [],
    content : [],
    content2 : [],
    user : [],
    bIdx : [],
    filename : []
  })

  // 페이지네이션 관련 관련 변수
  const [page, setPage] = useState(1);
  const [currentPost, setCurrentPost] = useState(board.title)
  const postPerPage = 6
  const indexOfLastPost = page * postPerPage
  const indexOfFirstPost = indexOfLastPost - postPerPage

  useEffect(()=>{
    axios.post("/boardloading",{
      message : 'boardloading'
    }).then((res)=> {
      console.log("react:",res)
      setBorad(...[
        title : res.data.title.reverse(),
        content : res.data.content.reverse(),
        content2 : res.data.content2.reverse(),
        user : res.data.user.reverse()
      ])
    })
  })
}
```

- useState를 사용해 게시판의 제목, 내용, 사용자 정보 등을 상태로 관리
- 현재 페이지와 게시물 수를 설정하고, 현재 페이지에 맞는 게시물 목록을 계산
- useEffect를 통해 컴포넌트 렌더링 시 서버에서 페이지 데이터를 로드하고 상태를 업데이트
- 데이터나 페이지 변경 시 업데이트

> 트러블 슈팅



1 문제

펫 정보 페이지에서 Pagination이 정상적으로 작동하지 않아 '다음' 버튼을 눌러도 목록이 갱신되지 않는 문제 발생



2 시도

Pagination과 펫 정보 목록의 상태(state) 불일치 문제를 해결하기 위해 Pagination의 초기 state 값을 설정



3 해결

map함수의 인덱스를 활용하여 데이터가 올바르게 갱신되도록 코드 수정하여 해결

> Contact



[입력문]
2가지가 있다.
사용자에게 직접 입력을 받는 방식과
사용자에게 확인을 입력 받는 방식이~

`input()`
적하세요, "ex)홍길동");
!창 위에 문구, 입력창에 작성문 예시 보여줌
!로 받는다

자동으로 인식해주고 오류도 최소화 시켜서 편리한 언어이긴한데
체계적으로 맞다 아니다 막 떨어지는 느낌이 아니라서 오히려 불편할 수 도 있음!

2. 사용자에게 확인 입력 -> `confirm()`



연락처

010-3191-6601

jvcki7@naver.com



GIT

<https://github.com/2024-SMHRD-IS-IOT-3/PP.git>

<https://github.com/2024-SMHRD-IS-IOT-3/GodRepo.git>



BLOG

<https://blog.naver.com/jvcki7/223437238323>