

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

- JavaScript, HTML, CSS, React, jQuery, Bootstrap
- Python, C++, Flask, Node.js, RestFul API
- SQL, Oracle, MySQL

> Project

딥러닝 기반 생체신호 분석 및 부정맥 검출 앱 서비스

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

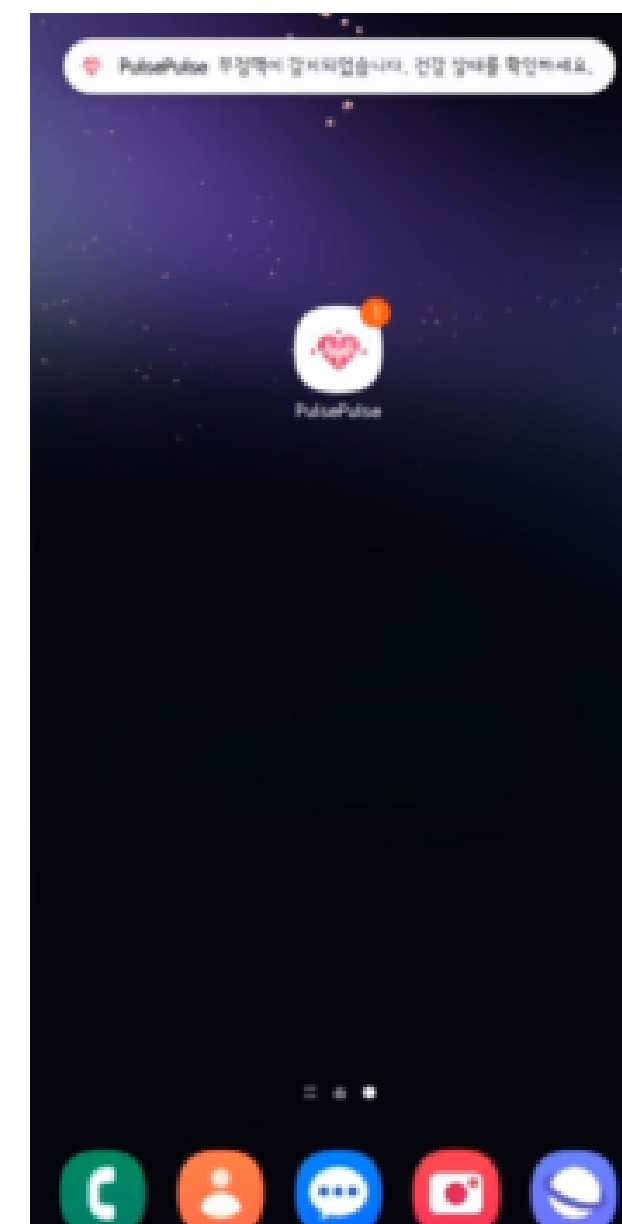


시작 화면

분석 결과 화면

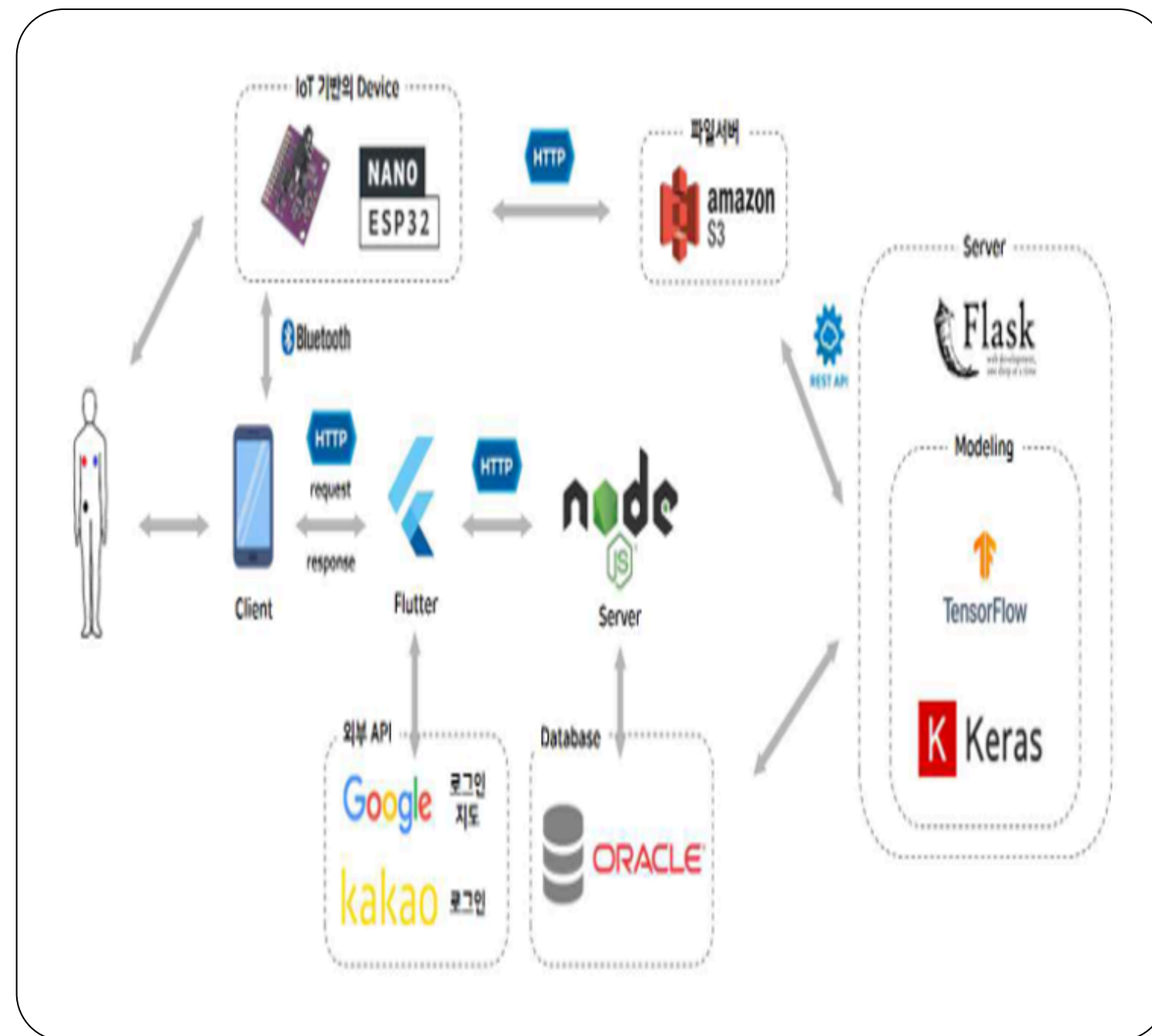


실시간 알림 화면

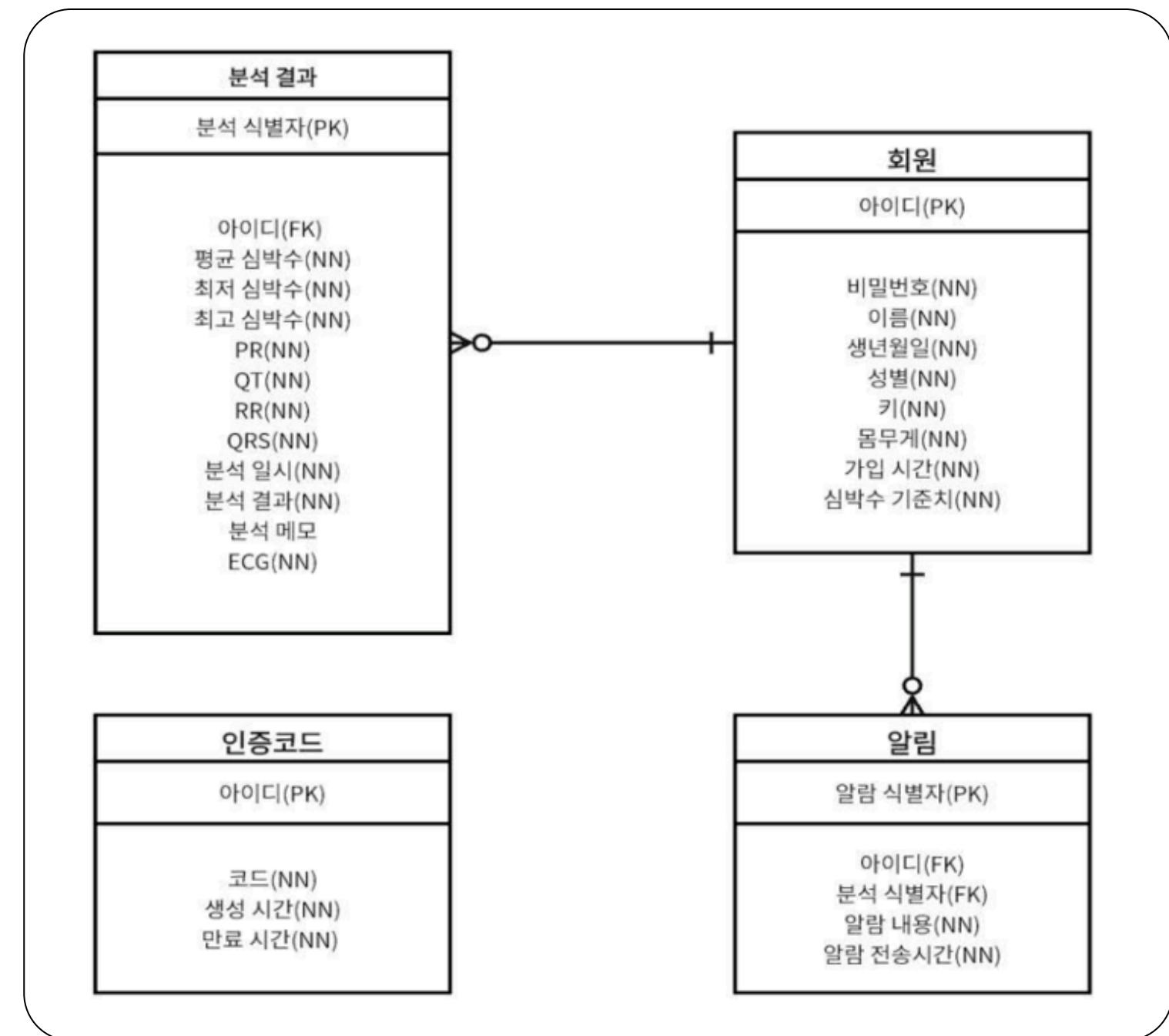


> Env & ERD

개발 환경



ERD



> Build & Resolve

담당역할 : Back-End개발, DB관리, Front-End 개발



소셜 로그인



JWT 토큰화



PDF 다운로드



알림 기능



트러블 슈팅

> 소셜 로그인 기능

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

```
// 카카오 로그인 라우터
router.get('/kakao', logMiddleware, passport.authenticate('kakao'));

// 카카오 로그인 콜백 라우터
router.get('/kakao/callback', async (req, res, next) => {
  const { code } = req.query;
  console.log("code", req.query);

  try {
    // 카카오에 인증 코드를 사용하여 액세스 토큰 요청
    const tokenResponse = await axios.post(
      'https://kauth.kakao.com/oauth/token',
      null,
      {
        params: {
          grant_type: 'authorization_code',
          client_id: KAKAO_CLIENT_ID,
          redirect_uri: KAKAO_CALLBACK_URL,
          code: code,
        },
        headers: {
          'Content-Type': 'application/x-www-form-urlencoded',
        },
      },
    );

    const accessToken = tokenResponse.data.access_token;
    console.log(tokenResponse.data);

    // 액세스 토큰을 Flutter 앱에 전달
    res.redirect(`/?token=${accessToken}`);
  } catch (error) {
    console.error('Error fetching token from Kakao:', error);
    res.redirect('/');
  }
});
```

```
_controller.loadRequest(Uri.parse(rootUrl)); // 메인 페이지로 이동
return NavigationDecision.prevent;
```

```
await _launchURL(initialUrl); // Google OAuth URL을 기본 웹 브라우저에서 열기
```

- passport 모듈을 사용해 카카오 인증을 진행한 후 액세스 토큰을 요청하여 Flutter 앱에 전달

- 웹뷰에서 현재 네비게이션을 중단, 메인 페이지로 강제 리다이렉트

- Google OAuth URL을 기본 웹 브라우저에서 열어 로그인 절차를 진행

> PDF 다운로드 기능

데이터베이스의 CLOB 데이터를 문자열로 변환하고, BLOB 데이터를 Base64로 인코딩하여 문자열로 다시 디코딩함으로써 ECG 차트를 시각화하고, 분석 결과를 PDF로 생성기능 제공

```
// CLOB을 문자열로 변환하는 함수
async function convertClobAsString(lob) {
  return new Promise((resolve, reject) => {
    if (lob === null) {
      return resolve(null);
    }

    let clobString = "";

    lob.setEncoding("utf8");
```

- CLOB 데이터를 문자열로 변환 후 반환
- 오류 발생 시 Promise로 전달

```
// Base64 문자열을 소수점 3자리까지의 숫자 배열로 변환하는 함수
function decodeBase64ToNumberArray(base64String) {
  try {
    const buffer = Buffer.from(base64String, "base64");
    const floatArray = new Float64Array(
      buffer.buffer,
      buffer.byteOffset,
      buffer.length / Float64Array.BYTES_PER_ELEMENT
```

- BLOB 데이터를 Base64 문자열, 이를 Float64 배열로 변환 후 숫자 배열로 반환

```
const PDFDocument = require("pdfkit");
```

- PDF 파일을 생성하고 다운로드할 수 있는 기능을 구현

> 알림 기능

부정맥이 검출되면 서버에서 Firebase Cloud Messaging(FCM)을 통해 사용자에게 푸시 알림 전송
사용자는 실시간으로 알림을 받아 현재 상태를 기록할 수 있도록 메모 기능 구현

```
let serAccount = require("../config/push-notification-key.json");

admin.initializeApp({
  credential: admin.credential.cert(serAccount),
});

module.exports = admin;
```

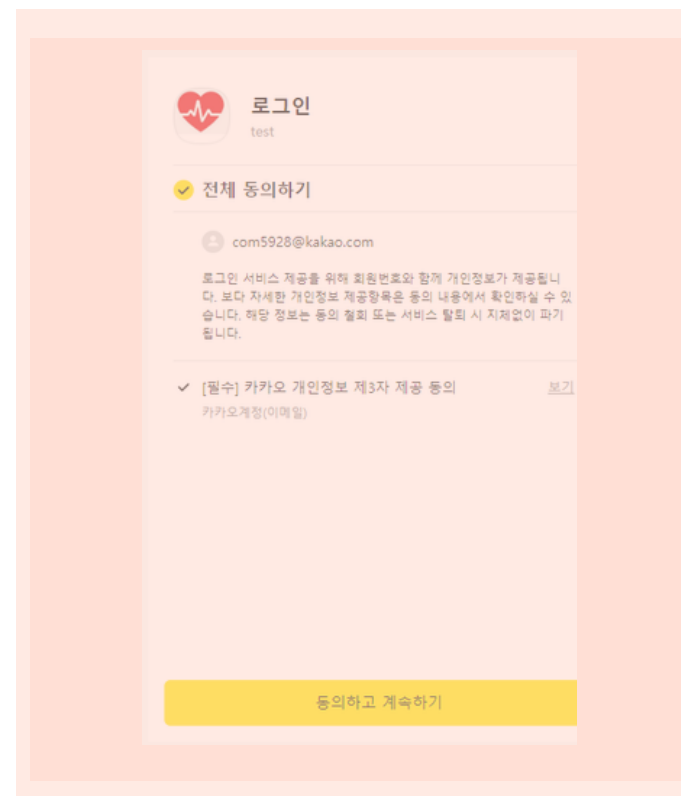
```
// 알림 트리거 메서드
Future<void> _showArrhythmiaNotification() async {
  // final bool hasVibrator = await Vibration.hasVibrator() ?? false;
  // if (hasVibrator) {
  //   Vibration.vibrate(duration: 1000); // 1초간 진동
  // }

  final AndroidNotificationDetails androidPlatformChannelSpecifics = AndroidNotif
    'arrhythmia_channel_id', // 채널 ID
    '부정맥 감지', // 채널 이름
    channelDescription: '부정맥 감지 알림을 위한 채널', // 채널 설명
    importance: Importance.max, // 중요도 설정
```

- 서버에서 Firebase Admin SDK를 초기화하여 Firebase 서비스에 인증된 접근을 설정

- 부정맥 감지 알림을 설정하고 표시하며, 테이블 상태에 따라 알림을 트리거할 수 있게 구현

> 트러블 슈팅

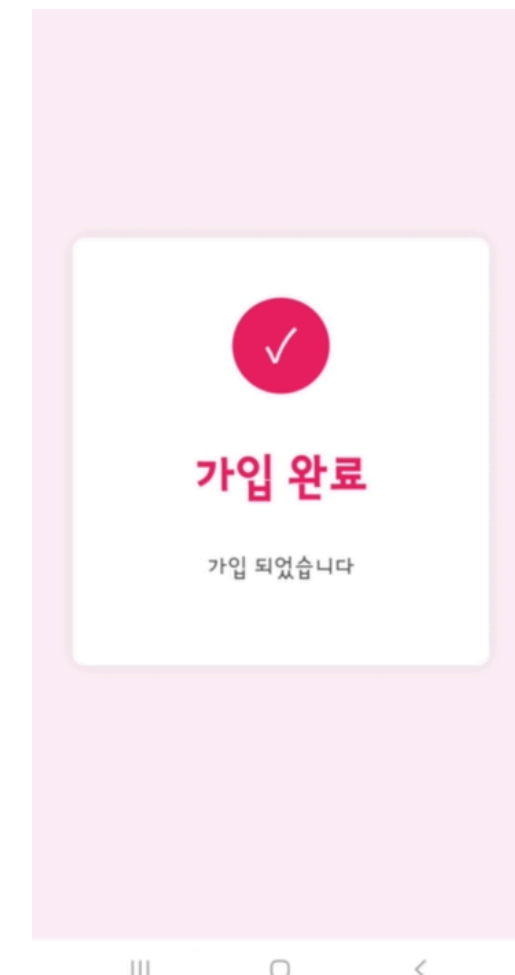


1 문제

Flutter 개발 환경과 소셜 로그인 SDK 간의 호환성 문제로 인증 오류 발생



```
dependencies:  
  webview_flutter: ^4.9.0
```



2 시도

Flutter의 WebView를 통해 Node.js 서버와 Restful API로 시도했으나, 구글 로그인이 인라인 WebView에서 작동하지 않음

3 해결

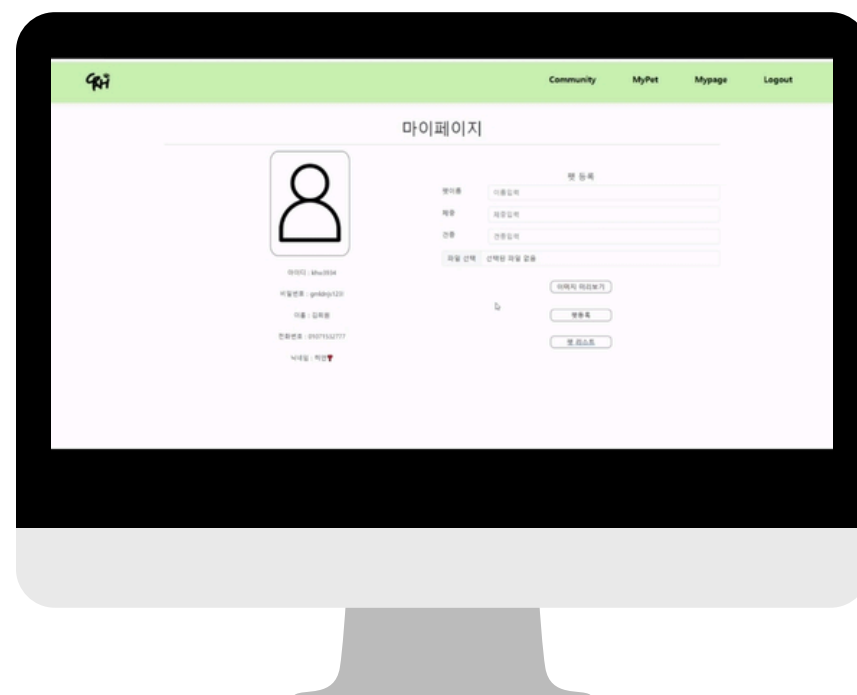
구글 로그인은 OAuth 인증을 외부 브라우저에서 처리하도록 변경, 카카오 로그인은 Node.js 서버에서 인증 후, 인라인 WebView에서 결과 화면을 표시

> Project

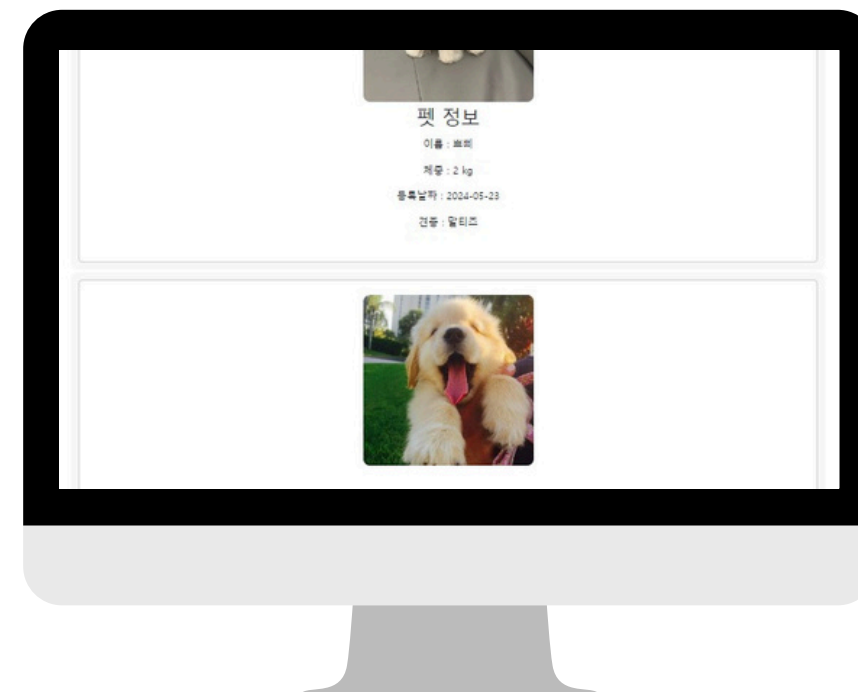
스마트홈 기술을 활용한 반려동물 케어 웹 서비스

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

회원 정보 페이지



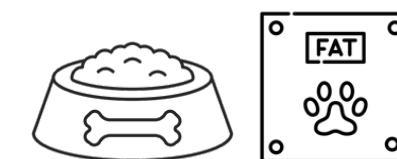
펫 정보 페이지



실시간 수집 데이터 관리

ORACLE

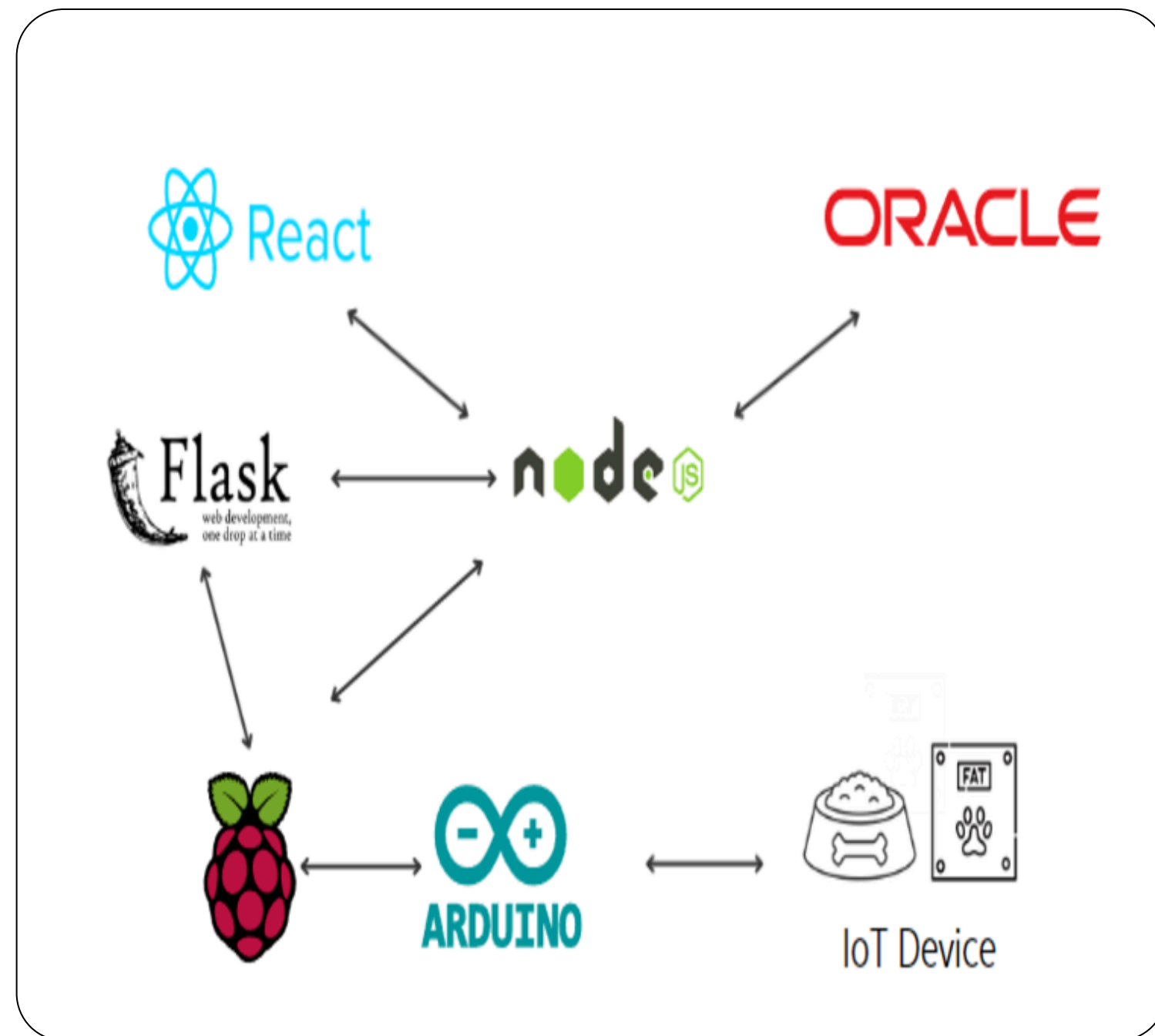
Flask
web development,
one drop at a time



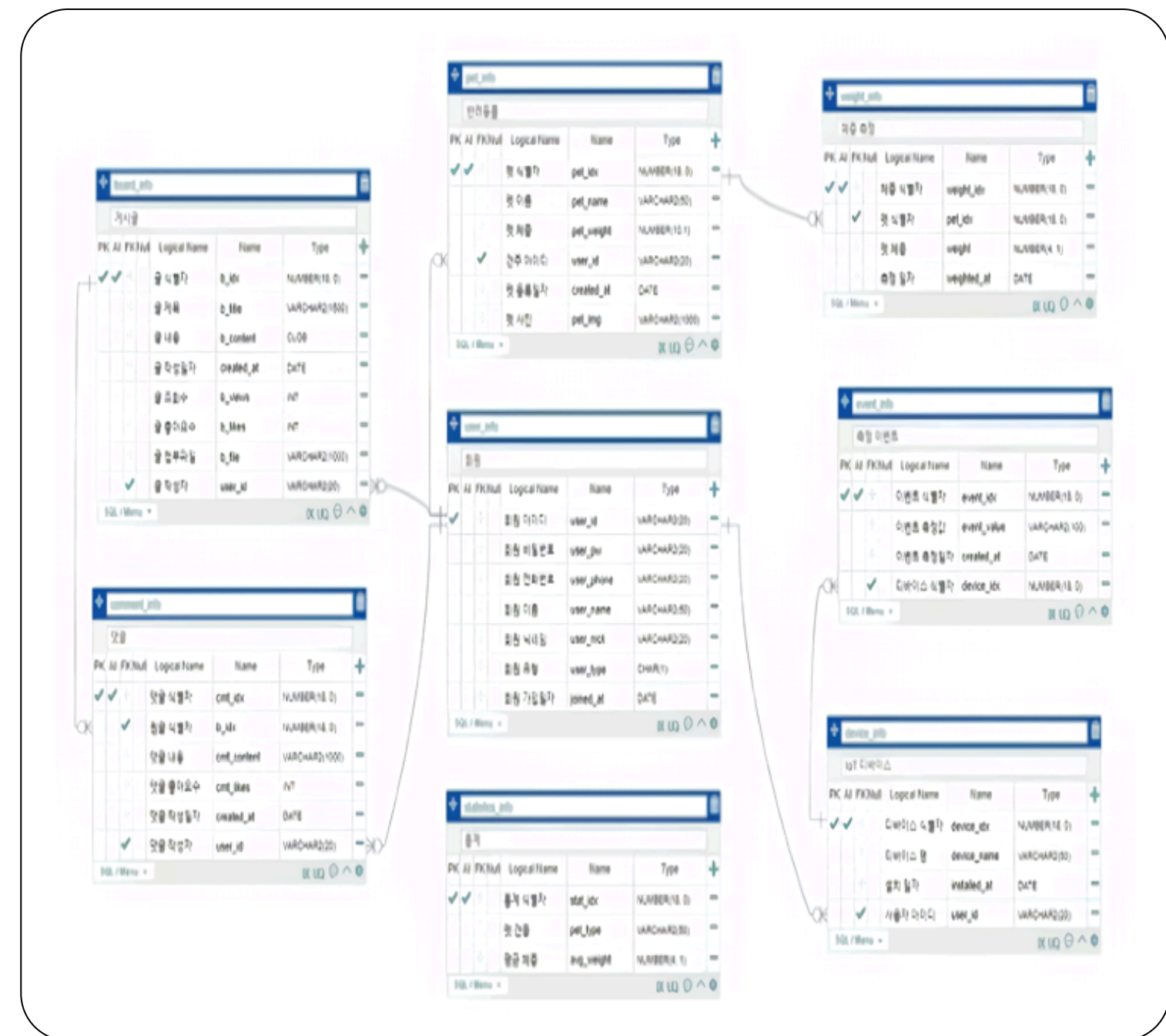
IoT Device

> Env & ERD

개발 환경



ERD

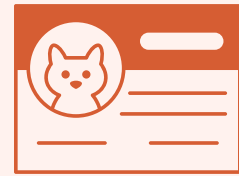


> Build & Resolve

담당역할 : Back-End개발, DB관리, Front-End 개발



회원 정보 관리



펫 정보 관리



실시간 데이터
관리



트러블 슈팅

> 회원 정보 관리

사용자 정보 관리 기능을 구현하고, 해당 데이터를 효율적으로 저장 및 관리하기 위해 스토리지 기능을 도입

```
{sessionStorage.getItem("user") && (  
  <div>  
  
  <ul>  
    <li>  
      <h5>회원정보</h5>  
      <hr />  
        
      <hr />  
    </li>  
  </ul>  
</div>  
)
```

- 로그인한 사용자의 정보를 sessionStorage에서 가져와 화면에 표시하며, 사용자 정보가 있을 때만 렌더링
- 스타일이 적용된 카드 컴포넌트로 UI를 구성

> 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 아니어서 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>