

# PETRADAR PROJECT

---

길 잃은 반려동물을 빠르게 찾을 수 있는 위치 기반 반려동물 실종 신고 서비스 플랫폼

2025-06-16 ~ 2025-06-25

김아영, 임유림, 권택준, 남궁찬

# 목차

- 01 ━━━━━━ 프로젝트 개요
- 02 ━━━━━━ 프로젝트 팀 구성 및 역할
- 03 ━━━━━━ 프로젝트 수행 절차 및 방법
- 04 ━━━━━━ 프로젝트 수행 경과
- 05 ━━━━━━ 마케팅 전략 및 기업 이익 측면
- 06 ━━━━━━ 자체 평가 및 개인 의견
- 07 ━━━━━━ Q&A

# **프로젝트 개요**

# **프로젝트 주제**

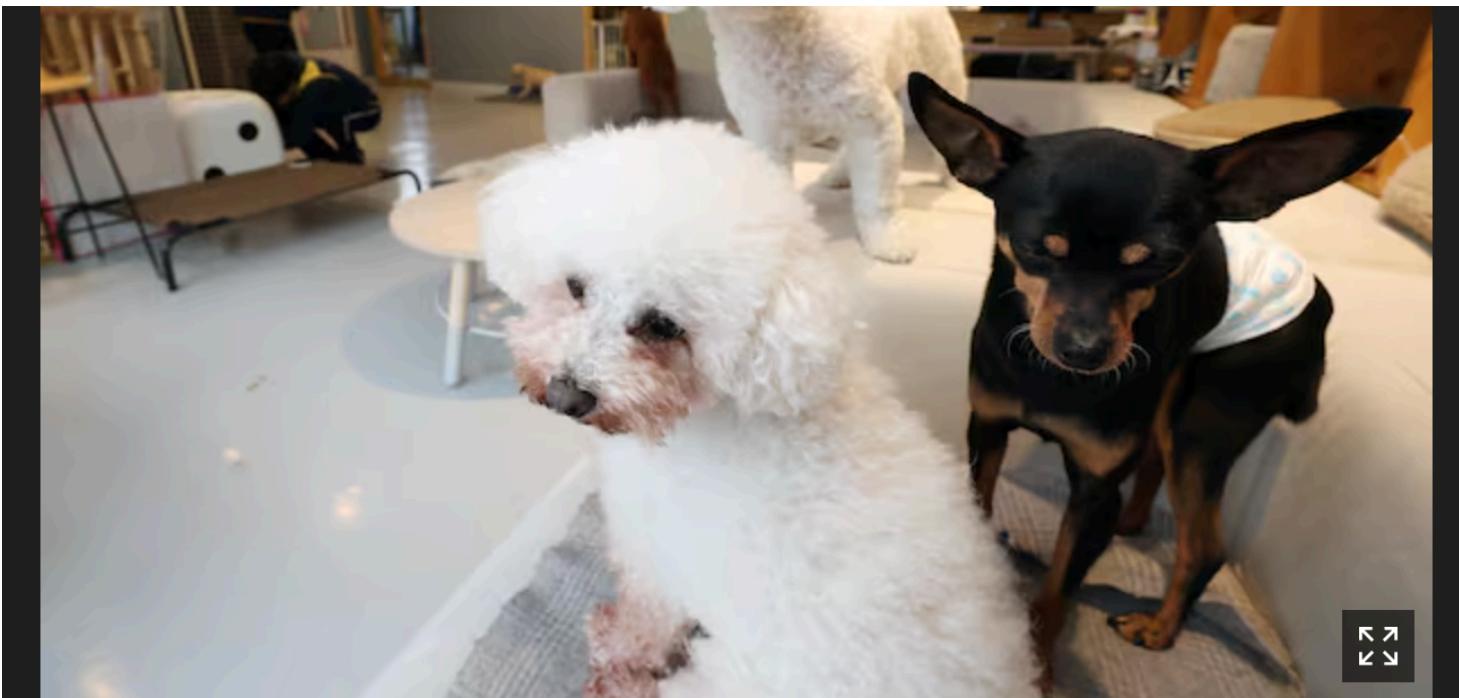
반려동물에게도 ‘골든타임’이 있습니다.

PetRadar는 실종된 반려동물을 더 빠르게, 더 넓게 알리기 위한 **위치 기반 반려동물 실종 신고 서비스 플랫폼**입니다.

**위치 등록, 실시간 알림, 제보 공유 기능**을 통해 반려동물 실종 시

빠르고 효율적으로 대응할 수 있도록 돋는 웹 기반 구조화 서비스입니다.

# 문제 정의 및 배경



지난해 설 연휴 때 버려진 유기견들이 서울 마포구 카라 더블어枢센터에서 입양을 기다리고 있다. 반려동물 인구 1500만명 시대지만, 전국 공공-사설 보호소에 들어오는 유기동물은 연 20만~30만 마리로 추산된다. /뉴스1

출처: 조선일보

## 유기·실종 증가

반려동물 실종·유기 문제는 매년 10만 마리가 넘는 규모  
**하루 유기/실종은 75마리 이상으로 심각**

정이 계속해서 증가하고 있다"며 "특히 실종된 반려동물들은 제온 유시가 어려워 생존이 더욱 위험해진다. 연합회는 전국 보육원과 협력하여 구조된 동물들의 신속한 보...

한국목재신문 · 2025.02.11.

## 고양이·강아지 실종 증가... 한국동물보육원연합회, 탐정 활동 강화

또한 실종된 동물의 이동 패턴을 분석하는 '반려동물 탐정팀'을 운영해 보다 체계적인 수색을 진행하고 있다. 관계자는 "고양이는 겁이 많아 좁고 어두운 곳에 숨는 경향이 있고, 강아지는 사람을 따라 이동하는 경우가 많다"며 "이러한 특성을 고려해 수색 ...

지피코리아 · 2025.02.28.

[한국동물보육원연합회, 강아지·고양이 탐정 서비스 신뢰 구축 앞장](#)

출처: 한국목재신문

## 낮은 재회율

보호소 입소 동물 중 **단 10-20%만이 재회**  
또는 입양됨이 일반적인 추정치

# 문제 정의 및 배경

01

실종 시 대부분 보호자들은  
**SNS, 포스터, 커뮤니티에 의존**

02

**실시간 전파력, 수색 범위가 한정**  
되어 평균 7일 이상 소요

03

**재회율이 낮고,**  
많은 동물들이 구조되지 못함

# 프로젝트 목표

01

## 실종 후 ‘골든타임’을 지키는 빠른 정보 공유

반려동물 실종 직후, 보호자는 정보 확산에 어려움을 겪습니다. 이를 해결하기 위해, 위치 기반 실시간 정보 등록 및 알림 기능을 통해 신속하게 주변 사용자에게 실종 사실을 전파하는 것을 목표로 합니다.

02

## 지역 커뮤니티의 힘을 활용한 자발적 구조 협력

실종 동물 수색은 보호자만의 힘으로는 어려운 경우가 많습니다. PetRadar는 **지도 기반 제보 등록, 사진 포함 공유, 제보자 참여 유도**를 통해 다수의 사용자 참여 기반 구조 시스템을 실현합니다.

“반려동물에게도 ‘골든타임’이 중요합니다.”

빠른 정보 공유 → 빠른 수색 → 빠른 재회. PetRadar는 이 모든 과정을 연결해주는 구조 플랫폼입니다.

# 타겟 지역 및 사용자

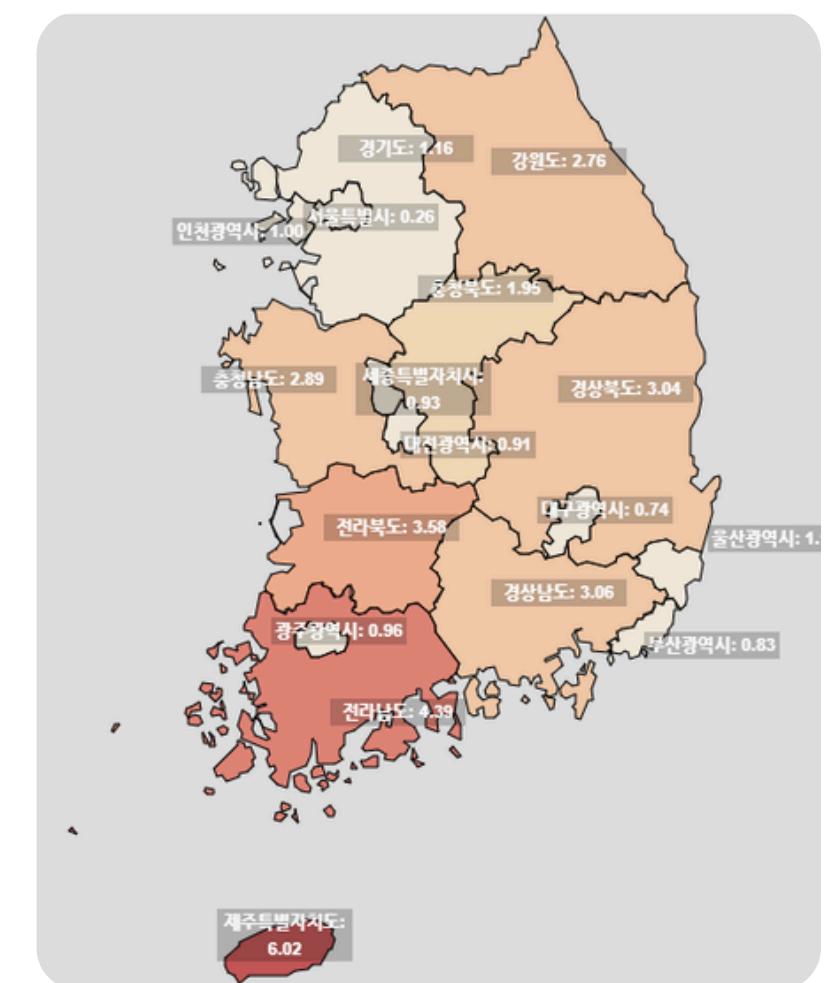
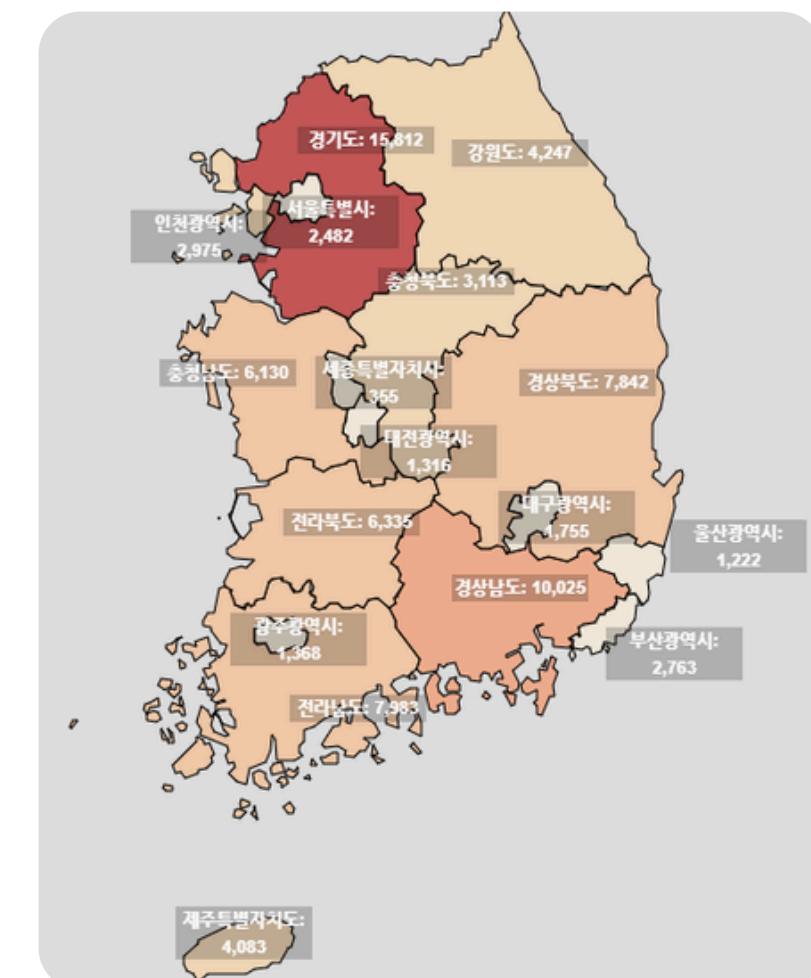
## 광역자치단체별 유기건수

지역별 유기 건수를 분석한 결과,

경기도에서는 15,812건으로 가장 많은 동물 유기가 발생

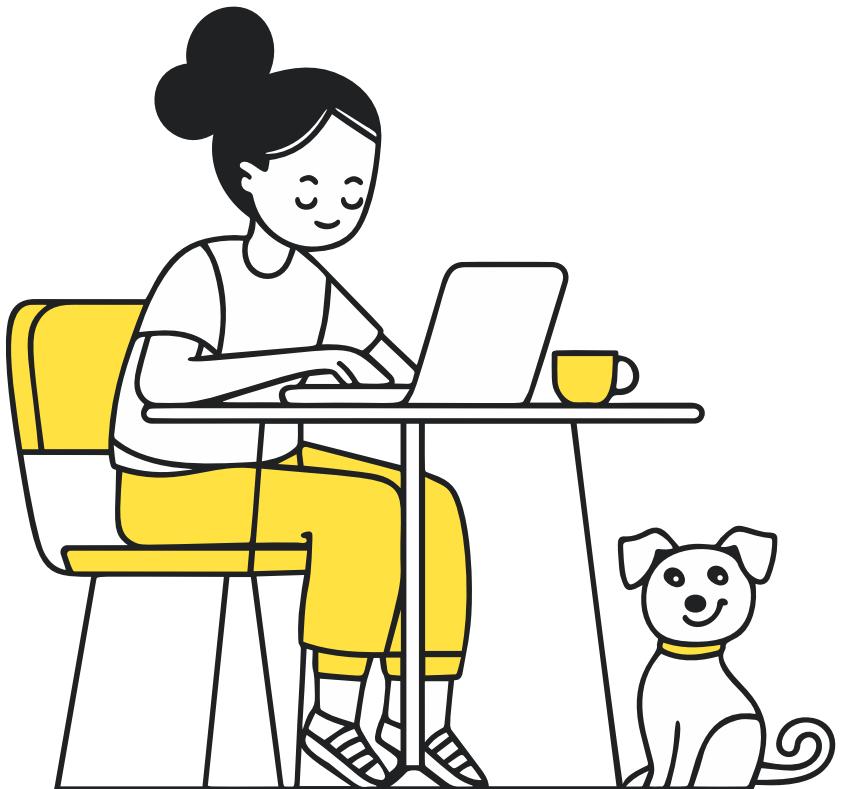
전국 유기동물의 약 40%가 보호나 입양 없이 남고,

강원도 역시 지자체 보호소 과포화 우려



출처: 농림축산식품 공공데이터 포털

# 사용자 페르소나



김수진 32세 직장인

## 거주지

경기도 양주시

## 거주형태

1인 가구, 반려견과 생활

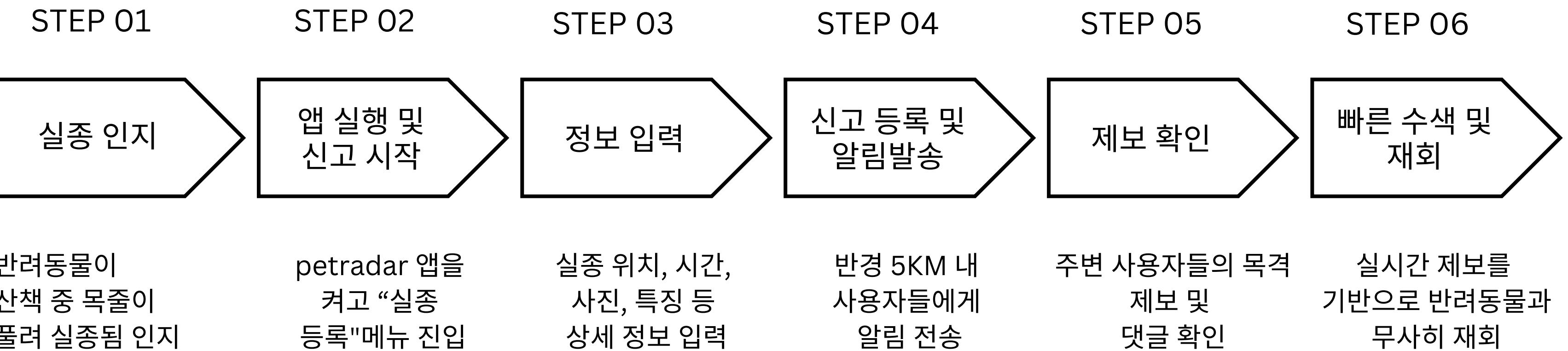
## 행동 특성

반려견을 가족처럼 아끼며, 실종 등 위기 상황에 민감

## Needs

산책 중 돌발 상황 대비 수단 필요  
실종 시 빠른 신고 및 주변 도움 연결

# 사용자 시나리오



# **프로젝트 팀 구성 및 역할**

# 팀 구성 및 역할



김아영

레이아웃 구조, 메인페이지,  
디자인, 프로젝트 총괄, css,  
인원 파트 분배 및 프로젝트 지휘



임유림

실종/제보 등록 및 목록 기능 담당  
Input 입력폼 구현 및  
카카오맵 API



권택준

사용자 인증(회원가입/로그인),  
알람 담당, 데이터구조  
마이페이지 카카오맵 API



남궁찬

보호소 정보 연계 및 시각화 담당,  
보호소 정보 공공데이터 API,  
위치기반 카카오맵 API

# **프로젝트 수행 절차 및 방법**

# 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	6/16(월) ~ 6/17(화)	<ul style="list-style-type: none"><li>· 프로젝트 기획 및 주제 선정</li><li>· 기획안 작성</li><li>· 기술 스택 및 와이어프레임 정의</li></ul>	지도 기반 실종 동물 서비스 설계
데이터 수집	6/17(화)	<ul style="list-style-type: none"><li>· 데이터 수집</li><li>· 기존 서비스 분석 및 차별점 도출</li><li>· API 데이터 스팩 파악 및 실현 가능성 파악</li></ul>	Kakao/경기도 보호소 API 스펙 분석
데이터 전처리	6/17(화) ~ 6/18(수)	<ul style="list-style-type: none"><li>· 데이터 정제</li><li>· 데이터 정규화</li><li>· API 필요 데이터 추출</li><li>· 웹스토리지 데이터 저장 포맷 정의</li></ul>	보호소 위치·입소일 등 필드 추출
모델링	6/18(수) ~ 6/19(목)	<ul style="list-style-type: none"><li>· 상태 관리 구조 설계 및 모델링</li><li>· UI/UX 모델링</li><li>· 페이지 라우팅 구조 외 정의</li><li>· 힙업 구조 설계</li></ul>	Context + Reducer, Git 브랜치 전략
서비스 구축	6/19(목) ~ 6/24(화)	<ul style="list-style-type: none"><li>· 플랫폼 기능 구현</li><li>· 실종동물 신고/제보</li><li>· 지도 기반 보호소 조회</li><li>· 알림 기능 (신고/제보 탐지) 및 로그인/로그아웃 구현</li><li>· 최종 기능 점검 (브라우저/모바일 환경 확인)</li></ul>	모바일/PC 테스트
총 개발기간	6/25(수)	<ul style="list-style-type: none"><li>· 전체 기간 종합 개발</li><li>· 배포(vercel) 예정</li></ul>	총 10일 개발 일정

# **프로젝트 수행 경과**

# 기술스택

## React

공통 컴포넌트 재사용  
**React Router**  
페이지 분리  
**Context + Reducer 패턴**  
사용자, 실종글, 제보글 데이터 상태 관리  
**데이터 구조**

사용자(User)	실종글(Missing)	제보글(Report)
아이디(PK)	실종글 고유번호(PK)	제보글 고유번호(PK)
...	사용자 아이디(FK)	실종글 고유번호(FK)
	...	사용자 아이디(FK)
		...
	실종 위치	제보 위치

## GitHub

팀원 간 코드는 GitHub 브랜치 기반으로 관리하였으며,  
각 기능별 브랜치에서 개발 후 Pull Request를 통해  
병합을 진행했습니다.  
충돌 해결과 버그 수정, 이슈 트래킹을  
통해 협업 품질을 높였습니다.

<https://github.com/ahyoung611/PetRadar.git>

## Vercel

Vercel은 React로 구현한  
프론트엔드 코드를 빠르게 배포하기  
위한 플랫폼으로 활용하였습니다.  
GitHub와 연동하여 기능 개발 중 자동  
배포가 이루어졌습니다.

<https://pet-radar-ahyoungs-projects-6db0c3a9.vercel.app>

# 외부 API

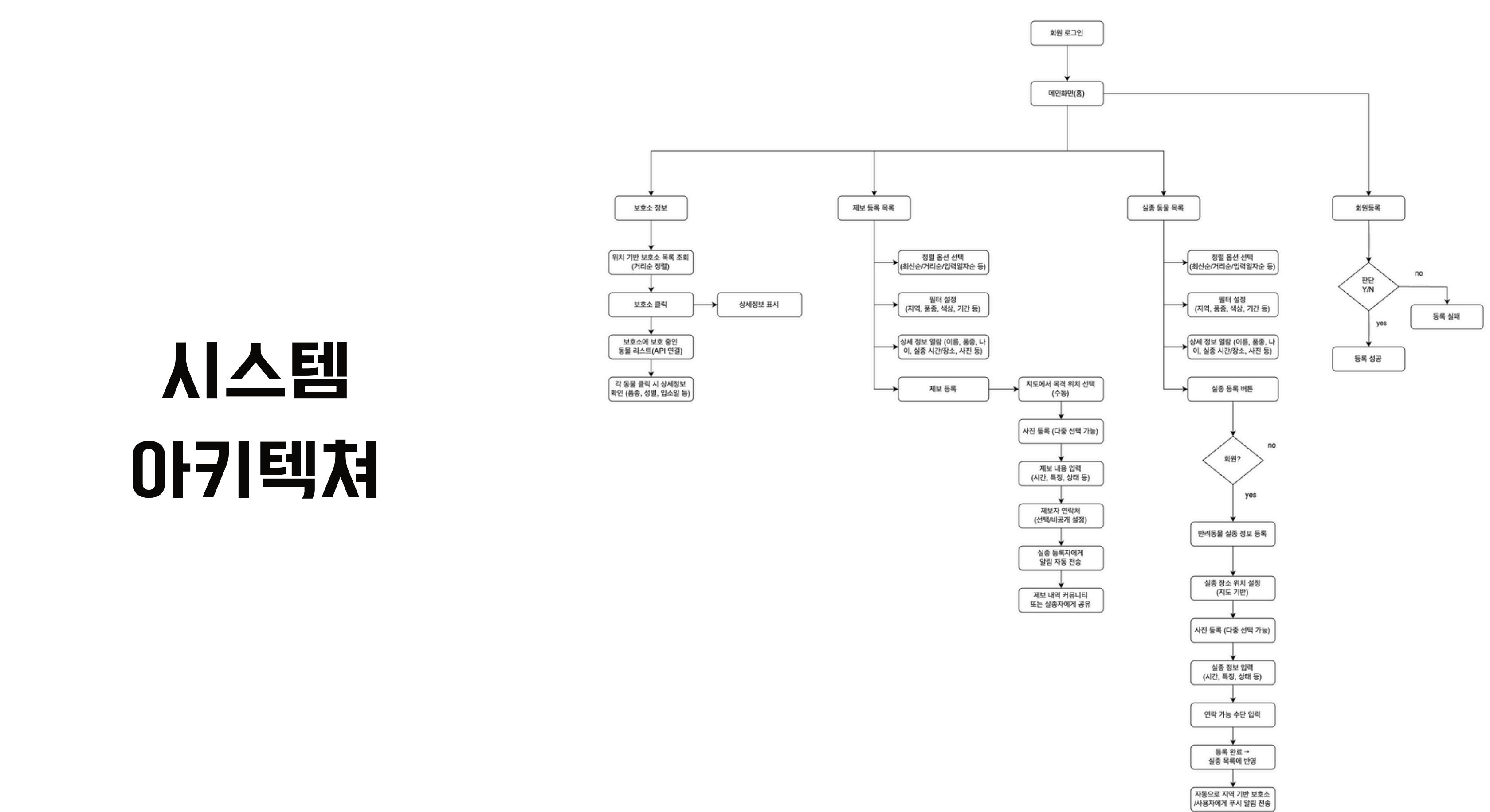
## 카카오맵 API

실종글과 제보글 데이터에는  
지도 API를 통해 받아온  
장소 좌표(위도, 경도)가 저장되며  
이를 기반으로 지도에서  
실종 및 제보 위치 시각화

## 공공데이터 보호소 API

Axios를 통해 보호소 데이터 연계  
사용자는 이 데이터를 통해 실종동물을  
보호소에서도 찾아볼 수 있다.

# 시스템 아키텍쳐



# 주요 기능

## 실종 동물 신고 등록 및 지도 연동

사용자가 실종된 반려동물의 정보를 등록하면  
Kakao Map 기반 지도에 마커로 표시됩니다.

## 실시간 알림 서비스

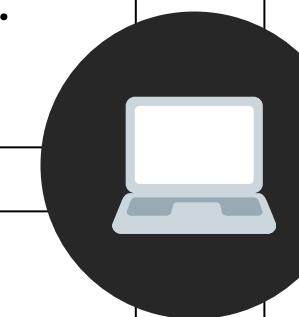
실종 신고가 등록되면 근처 사용자에게  
즉시 알림이 전송됩니다.

## 로그인/회원가입

회원가입 및 로그인 기능 구현

## 주민 인식 변화

실종 동물을 목격한 사용자가 사진, 위치,  
제보 내용을 입력하여 제보를 등록할 수 있습니다.



# 카카오맵 API를 이용한 지도 구축

## Home 페이지

실종 신고된 동물 위치 마커 전체 표기

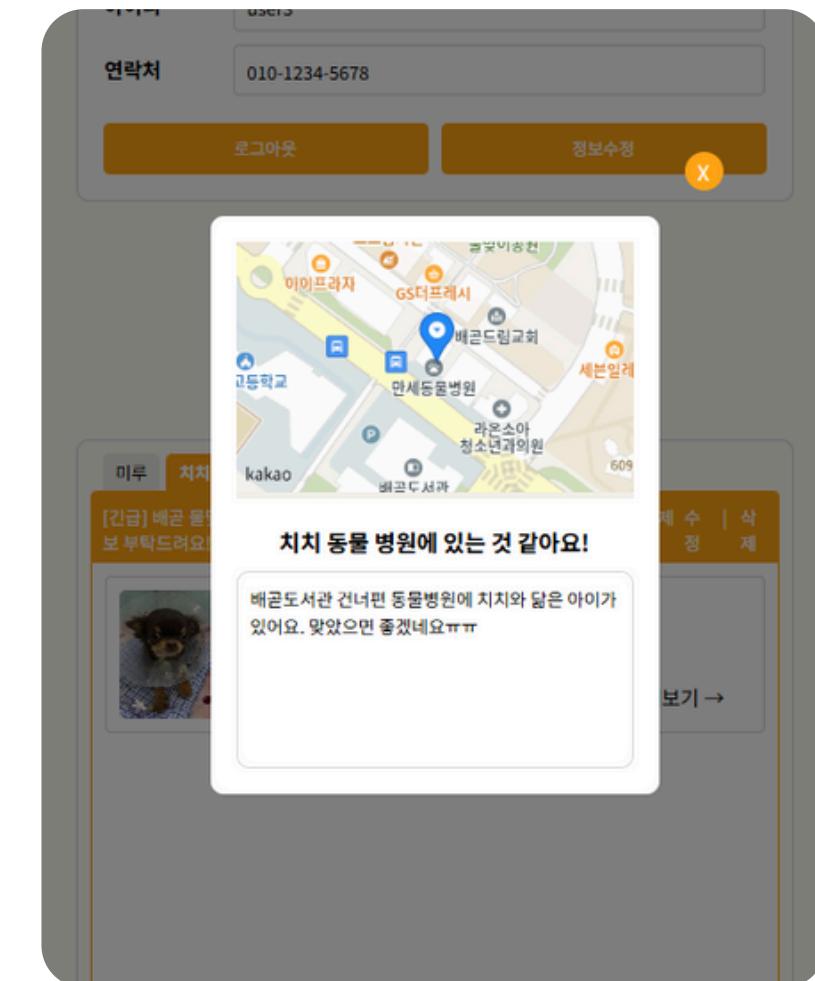
## MissingDeclaration, MissingReport 페이지

실종 신고/제보 시 실종 위치/발견 위치

사용자가 선택 후 context 저장

## MyPageReportMap

카카오맵 API Static Map 활용



```
MyPageReportMap.jsx 1 ×
components > MyPageReportMap.jsx > MyPageReportMap > initializeStaticMap > options
import { useEffect, useRef } from 'react';

const AppKey = "b737c6777956f74337fc9bc5a08e3b55";

const MyPageReportMap = ({ petReportPoint }) => {
  const mapRef = useRef(null);
  const staticMapRef = useRef(null);

  const initializeStaticMap = () => {
    if (mapRef.current) {
      mapRef.current.innerHTML = "";
    }
    const center = new window.kakao.maps.LatLng(
      petReportPoint.lat,
      petReportPoint.lng
    );
    const marker = { position: center };
    const options = [
      center,
      level: 3,
      marker
    ];
    staticMapRef.current = new window.kakao.maps.StaticMap(
      mapRef.current,
      options
    );
  };

  useEffect(() => {
    if (!petReportPoint) return;
    if (!window.kakao || !window.kakao.maps) {
      const script = document.createElement("script");
      script.src = "https://dapi.kakao.com/v2/maps/sdk.js?appkey=${AppKey}&libraries=services&autoload=false";
      script.async = true;
      script.onload = () => {
        window.kakao.maps.load(() => {
          initializeStaticMap();
        });
      };
      document.head.appendChild(script);
    } else {
      initializeStaticMap();
    }
  }, [petReportPoint]); // React Hook useEffect has a missing dependency: 'initializeStaticMap'. Either include it in the dependency array or remove the dependency.
  return <div ref={mapRef} style={{ width: "100%", height: "100%" }}></div>;
};

export default MyPageReportMap;
```

# 실시간 업데이트 알림 시스템

로그인한 유저라면 마지막으로 알림을

확인한 시각 이후에 등록된 나에게 온 제보글 데이터와

다른 유저들의 실종글 데이터를 조회

두 데이터를 병합한 뒤,

정렬하여 유저의 알림 목록에 추가

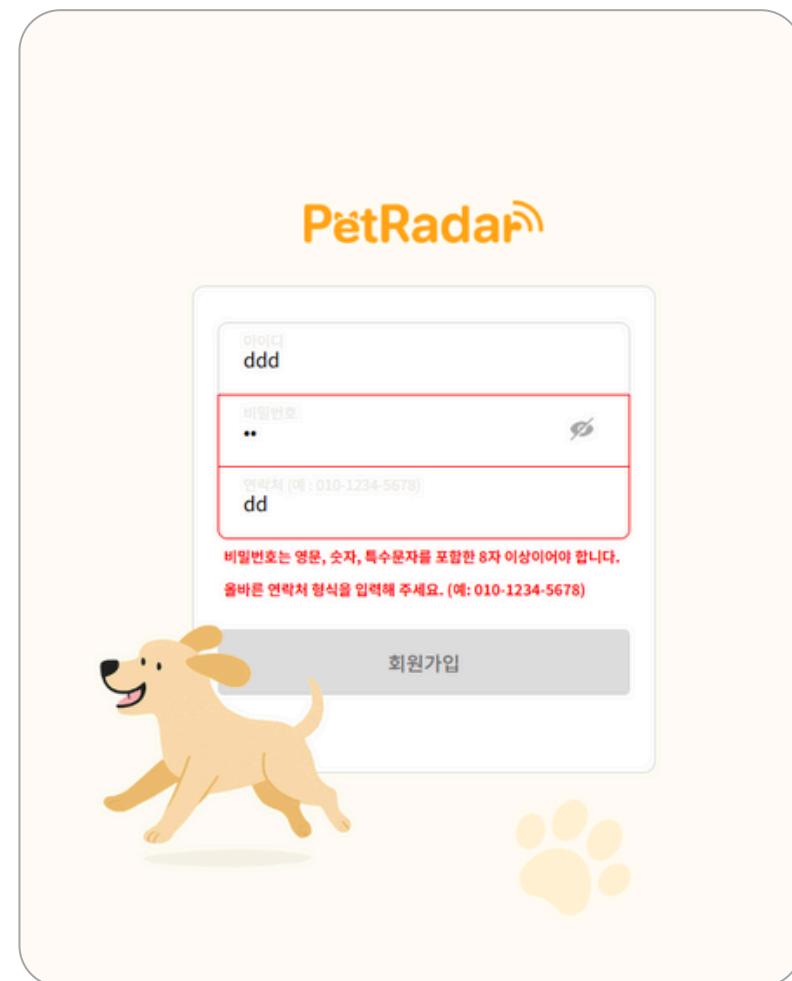
그 후에 마지막으로 알림을 확인한 시각 업데이트



```
const { isActive, toggleSidebar } = useSidebar();
const userState = useUserState();
const userDispatch = useUserDispatch();
const missingState = useMissingState();
const reportState = useReportState();
const userInfo = userState.users.find(
  (user) => user.id === userState.currentUser
);
useEffect(() => {
  let newAlert = [];
  if (userInfo) {
    const lastAlertDate = userInfo.lastAlertDate ?? userInfo.createDate ?? 0;
    const missingAlert = missingState
      .filter(
        (item) => lastAlertDate < item.createDate && item.id !== userInfo.id
      )
      .map((item) => ({
        id: item.id,
        postId: item.petMissingId,
        postType: "missing",
        createDate: item.createDate,
      }));
    const reportAlert = reportState
      .filter(
        (item) => lastAlertDate < item.createDate && item.petMissingUser === userInfo.id
      )
      .map((item) => ({
        id: item.id,
        postId: item.petReportId,
        postType: "report",
        createDate: item.createDate,
      }));
    newAlert = [...missingAlert, ...reportAlert].toSorted(
      (prev, next) => next.createDate - prev.createDate
    );
  }
  userDispatch({
    type: "ADD_ALERT",
    data: {
      id: userInfo.id,
      alerts: newAlert,
    },
  });
});
```

# 정규식을 사용한 회원가입 검증

회원가입 시 정규식 테스트를 이용해서  
입력 데이터 검증



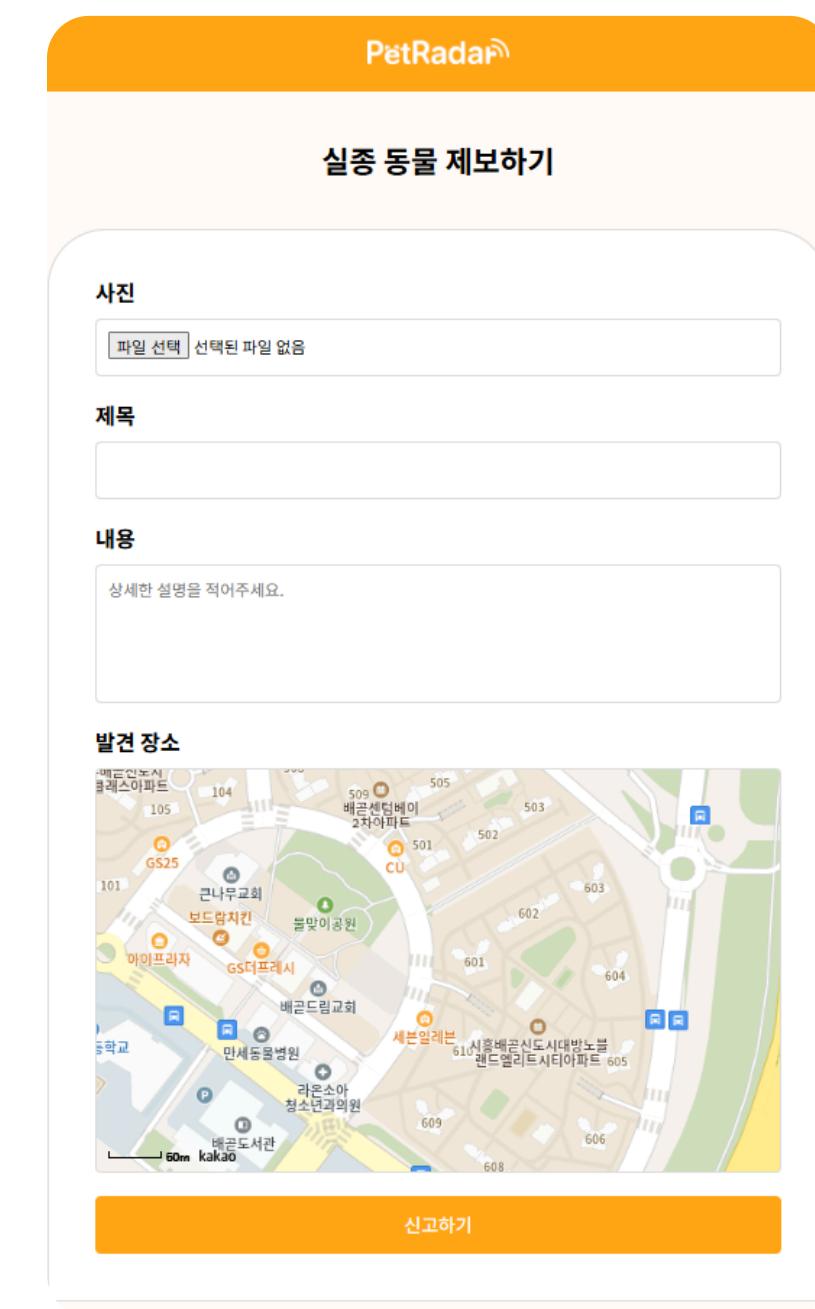
```
const formCheck = (name) => {
  const newErrMsg = {};
  const idRegex = /^[a-zA-Z]*$/;
  if (!input.id) {
    newErrMsg.id = "아이디를 입력해주세요.";
  } else if (isExist(input.id)) {
    newErrMsg.id = "사용할 수 없는 아이디입니다. 다른 아이디로 사용해주세요.";
  } else if (!idRegex.test(input.id)) {
    newErrMsg.id = "ID는 영문 소문자와 숫자만 입력 가능합니다.";
  }
  const pwRegex = /^(?=.*[a-zA-Z])(?=.*\d)(?=.*[^w\s]).{8,}/;
  if (!input.pw) {
    newErrMsg.pw = "비밀번호를 입력해주세요.";
  } else if (!pwRegex.test(input.pw)) {
    newErrMsg.pw =
      "비밀번호는 영문, 숫자, 특수문자를 포함한 8자 이상이어야 합니다.";
  }
  const hpRegex = /^01[0-9]{1}-\d{3,4}-\d{4}$/;
  if (!input.hp) {
    newErrMsg.hp = "연락처를 입력해주세요.";
  } else if (!hpRegex.test(input.hp)) {
    newErrMsg.hp = "올바른 연락처 형식을 입력해 주세요. (예: 010-1234-5678)";
  }
  setErrMsg(name ? { ...errMsg, [name]: newErrMsg[name] } : errMsg);
  return JSON.stringify(newErrMsg) === "{}";
};
```

# INPUT 태그 FORM 전송 및 FILEREADER 활용

# 실종 동물을 목격한 사용자의 입력 내용을 받아 실종 신고자에게 전달

# Input 태그에서 값을 받아 **target** 후 form으로 전송

이미지 파일의 경우 **FileReader** 객체를 생성하여  
**base64** 문자열로 반환하고 상태에 저장



```
const onCreate = () => {
  dispatch({
    type: "CREATE",
    data: {
      ...form,
      id: userState.currentUser,
      petMissingUser: currMissing.id,
      petMissingId: currMissing.petMissingId,
    },
  });
};

const onSubmitButtonClick = () => {...};
};

const handleChange = (e) => {
  const { name, value, files } = e.target;
  if (name === "petImage") {
    // 파일 미입력시 종료
    if (!files[0]) return;

    const reader = new FileReader();
    reader.onloadend = () => {
      setForm((prev) => ({
        ...prev,
        [name]: reader.result,
      }));
    };
    reader.readAsDataURL(files[0]);
  } else {
    setForm((prev) => ({
      ...prev,
      [name]: value,
    }));
  }
};

const onLocationSelect = (latlng) => [
  setForm((prev) => ({
    ...prev,
    petReportPoint: {
      lat: latlng.lat,
    }
  })),
];
```

# 시연 영상

시연 영상

# **마케팅 전략 및 기업 이익 측면**

# 마케팅 전략

01

## 사전 등록 유도

- **전략 목적:** 실종 발생 전, 사용자 데이터 확보 및 빠른 대응 구조 마련
- **구체 방안:** GPS 목걸이 연동 기능(예고): 반려동물 위치 추적 연동 시스템 준비, QR 코드 태그 등록 가능(예고): 미리 등록 시, 실종 시 자동 전파 가능
- **기대 효과:** 사용자 선등록 유도 → 빠른 신고 시스템 연계 → 보호 가능성 증가

02

## 감정 기반 UX 설계

- **전략 목적:** 보호자의 심리적 불안을 고려한 사용자 친화형 서비스 설계
- **구체 방안:** “반려동물에게도 골든타임이 존재합니다”와 같은 감성 문구 사용, 붉은 강조 색상, 울타리/이동 경로 등을 시각적으로 강조
- **기대 효과:** 사용자 공감 유도 → 서비스 신뢰성 향상 → 반복 이용 유도

# 마케팅 전략

03

## 커뮤니티 기반 구조 설계

- **전략 목적:** 사용자 제보 참여를 통해  
실질적인 실종 구조 확률 상승
- **구체 방안:** 실종/제보 글에 댓글 기능, 좋아요 기능  
탑재, “제보왕 뱃지”, “기여 포인트” 등  
참여자 보상 시스템 준비
- **기대 효과:** 커뮤니티 활성화 → 실종 구조율 상승  
→ 지속적 플랫폼 활성 유지

04

## SNS 공유 기능 강화

- **전략 목적:** 플랫폼 외부 확산  
→ 빠른 정보 전달과 사용자 유입
- **구체 방안:** 실종 상세 페이지 하단에 카카오톡/인  
스타그램/네이버 카페 공유 버튼 탑재, 공유 시 썸  
네일 이미지 + 간략 정보 자동 생성 기능 구현
- **기대 효과:** 실시간 지역 커뮤니티 공유  
→ 목격자 도달 확률 증가

# 기업 이익 측면

01

## 프리미엄 실종 등록 서비스 (부분 유료화 모델)

- 내용: 사용자가 일정 금액을 지불하면  
실종 반려동물의 게시물이 메인화면  
**상단 또는 지도에서 강조 노출**
- 예시 기능: 실종 등록 상단 고정,  
지도 마커 확대 및 컬러 강조, 긴급 배너 표시
- 수익 모델: 기본 등록은 무료, 프리미엄 노출 옵션  
(3일간 5,000원 / 7일간 9,000원 등)

02

## 반려동물 관련 업종과의 광고 제휴

- 대상 업종: 동물병원, 미용샵, 펫보험사,  
펫용품 쇼핑몰, GPS 위치 추적기업체 등
- 운영 방식: 실종 반려동물 위치 주변 병원 정보를  
노출하며 광고와 병행, 앱 내 제휴 배너, 푸시 광고,  
**위치 기반 광고 삽입**
- 수익 구조: 클릭 기반 광고 수익, 제휴 가입 시  
수수료 정산 (예: 보험 가입 시 10% 정산 등)

# 기업 이익 측면

03

## B2B 시스템 공급 (지자체·보호소 대상)

- **기능 제안:** 보호소/지자체 시스템과 연동하여  
실종 등록 시 자동 통보, 유기동물 입소 시스템과  
실종 제보 자동 매칭
- **수익 구조:** 지자체 대상 시스템 도입 비용 청구,  
유지보수 계약을 통한 정기 수익 발생

04

## 사례금 분배 시스템 도입 제안

- **기능 개요:** 제보를 통해 실종 동물의 재회에 기여한  
사용자에게 포인트 혹은 현금성 보상 제공
- **운영 구조:** 보호자가 '사례금 설정' 등록,  
제보자가 제보 후 '재회 성공' 처리 시 보상금 지급
- **이익 관점:** 플랫폼 수수료 (예: 사례금의 10~15%)  
수익, 사용자 참여 동기 강화로  
서비스 체류 시간 및 재방문율 상승

# **자체 평가 및 개인 의견**

# IID백 및 업그레이드 방안

기능 개선 필요 요소	기술적 제약 사항	업그레이드	대안	향후 확장 방향
사용자 GPS 위치 기반 지도 서비스	· 사용자 위치를 기반으로 실종/체포 알림 제공	· 로컬 환경에서 HTTP 기반 실행 시 브라우저 위치 권한 요청이 제한됨	· 사용자 및 보호소의 위치를 고정값으로 설정하여 기능 구현	· 앱 기반 구현 시 GPS 권한 요청을 통해 정밀한 위치 기반 서비스 구현
	· 사용자 주변 보호소 정보 안내	· 데스크탑 환경에서는 위치 오차가 커 정확도 확보 어려움		
실종 동물 보호소 연계 서비스	· 동물 보호소와 연계하여 실종 동물과 유사한 특징을 가진 개체 식별	· 경기도 지역 보호소 API만 활용 가능하여 지역 커버리지에 제한 존재	· 보호소 API를 활용하여 사용자의 위치 기반으로 인근 보호소 목록 조회 기능 제공	· 다양한 보호소 API 통합 검토 · GPS 목걸이/내장칩과의 연동으로 보호소 자동 매칭 고도화
		· 실종 동물과 보호 동물 간의 특징점 일치 여부를 판단하는 기준이 모호함		

# 추가 업그레이드 방안

## AI 기반 실종 탐색 추천 시스템

실종 지역 및 시간 데이터를 기반으로,  
유사 사례와 제보 패턴을 분석하여  
탐색 권역 및 시간대를 추천하는 기능  
→ 제보의 정확도와 구조 성공률 증가

## 반려동물 ‘사전 등록’ 관리 기능

사용자가 사전에 반려동물 정보를 등록하면,  
실종 시 자동으로 일부 정보가  
신고 양식에 불러와지는 기능  
→ 신고 시 스트레스 완화 및 빠른 대응 가능

## ‘구조 상황 업데이트’ 기능

신고 후 일정 시간이 지나면 보호자에게  
구조 상황 업데이트 유도 알림 전송  
→ ‘해결됨/해결 안됨’ 상태 분리로 신뢰도 강화

## 비상 연락 기능 / 핫라인 자동 생성

실종 후 보호자의 연락처가 자동으로 마커에  
노출되거나, 비공개 ‘긴급 연락하기’ 버튼 제공  
→ 즉시 구조자와 연락 가능, 실제 재회율 증가에 기여

# 느낀점 및 성과

김아영

PetRadar 프로젝트를 통해, 실제 사용자의 행동 흐름을 고려해 기획하고, 일관성 있는 디자인과 반응형 UI 구현에 집중하면서 디자인과 개발 사이의 연결 고리를 직접 경험할 수 있었습니다. 특히 styled-components를 활용해 스타일 시스템을 구조화하고, 다양한 디바이스에서 자연스럽게 동작하는 반응형 웹을 설계한 점이 큰 배움이었습니다. 다만, 로직이나 기능 구현보다는 UI와 CSS 중심의 역할을 맡았기에 프론트엔드 기능 개발에 깊이 참여하지 못한 점은 아쉬움으로 남습니다.

그럼에도 불구하고 팀원들과의 활발한 소통과 역할 분담을 통해 더 완성도 높은 결과물을 만들어냈다는 점은 의미 있는 성과였습니다.

임유림

PetRadar 프로젝트를 통해 실제 데이터를 다루고 사용자의 입장에서 생각하고 실용적인 방향성을 고려하며 개발하는 과정에 참여할 수 있어 의미 있었습니다.

API 연동과 로컬 스토리지를 활용한 새로운 방식은 익숙하지 않았지만, 직접 구현하며 배움의 즐거움을 느꼈고, 협업 과정 속에서 팀워크와 계획의 중요성도 자연스럽게 체감할 수 있었습니다.

완성도 있는 결과물을 위해 함께 노력해가는 경험을 통해 성취감을 얻을 수 있었습니다.

# 느낀점 및 성과

권택준

펫레이더 프로젝트를 통해 단순히 기능을 구현하는 것을 넘어, 실제 사회 문제(반려동물 실종)에 기여할 수 있는 서비스 개발에 참여했다는 점에서 큰 보람을 느꼈습니다.  
무엇보다도 초기 단계에서 비어 있는 Git 저장소를 보며 느꼈던 막막함 속에서, 하나하나 기능을 추가하고 병합할 때마다 프로젝트가 완성되어 가는 것을 체감한 순간이 가장 큰 보람이었습니다.  
팀원들의 코드가 하나로 모여 점점 완성된 형태로 변해가는 모습을 보며 성취감과 협업의 힘을 깊이 느낄 수 있었습니다.

남궁찬

이번 기회를 통해 API를 연계해 각종 정보를 가져와 팀 프로젝트 내에서 이용하고 찾아보면서 알게 된 크롤링 또는 AI까지 연계시키면 학습된 모델을 이용해 ex)웹 중고 쇼핑사이트 같은 물자 물류 재무 시스템 구축 시에 빠른 카테고리, 상품 가격 등을 분류하고 소비자에게 최적화된 정보를 표출 할 수 있고 판매자 또한 빠른 절차처리를 할 수 있을 것이라는 느낌을 느꼈습니다.

# **Q&A**

궁금한 점 질문해 주세요.

# **THANK YOU**

**김아영, 임유림, 권택준, 남궁찬**