



반려견 출장 케어 서비스 플랫폼



TEAM PROJECT

24.09.02 ~ 10.18

01

OVERVIEW

기획 배경 및 목표



02

ARCHITECTURE

03

DATABASE

04

CODE

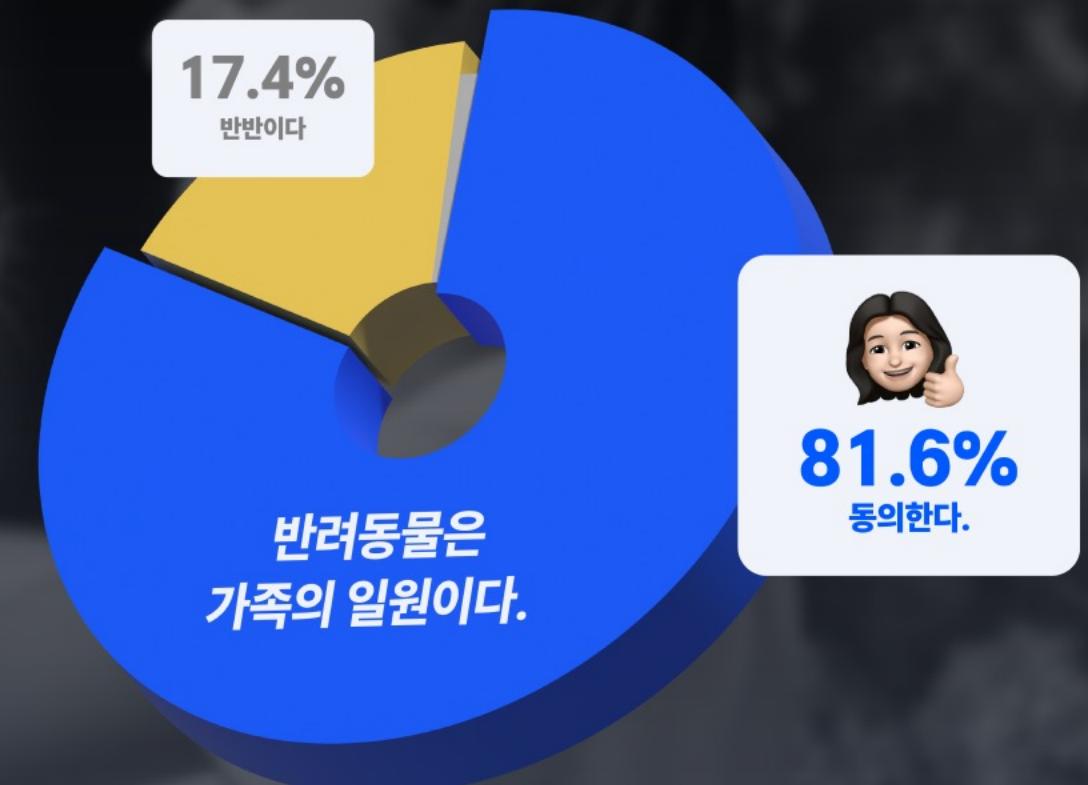
05

DEMONSTRATION

BACKGROUND

반려가구가 500만을 돌파함에 따른 **펫팸(Fet + Family)족의 증가**

펫팸족은 반려동물을 가족처럼 여기는 사람들을 이르는 단어로,
함께 살아가는 소중한 존재로 생각하는 이들이 늘어남에 따라
관련 경제산업인 ‘펫코노미’ 산업 또한 매년 다방면으로
급성장을 기록하고 있는 추세입니다.



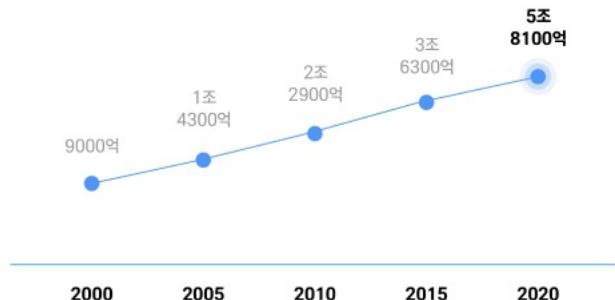
출처 : KB금융지주 2023 한국반려동물 보고서

펫코노미 산업 속 가장 주목을 받는 ‘서비스’와 ‘헬스케어’

꾸준히 관심도가 높았던 미용, 여행, 숙박 등 다양한 분야를 포함하는 ‘펫 서비스’와 반려견과 함께 더 많은 시간을 보내기 위해 급부상한 ‘펫 헬스케어’에 초점을 두었습니다.

DESKRESEACH ①

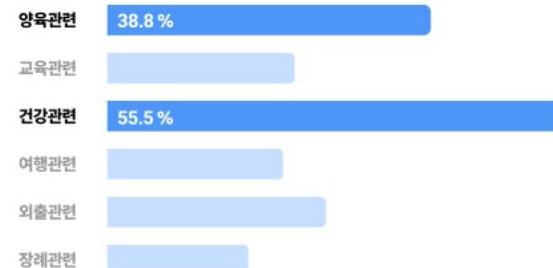
반려동물 시장 규모 변화



출처 : 1인 가구의 반려견 분리불안 완화를 위한 제품디자인 제안, Vol.5 No.2

DESKRESEACH ②

반려동물 관련 최근 관심사



출처 : KB금융지주 2023 한국반려동물 보고서

펫 푸드

펫 보험

반려동물 호텔링

펫 휴머나제이션

반려동물 장례 서비스

강아지 유치원

펫 헬스케어

펫 택시

동반 비행 티켓

POINT

미용 및 목욕, 건강과 같은 기본 케어에 주목했을 때,

기존 서비스에서 **불편한 점**이 있다면?

1

스케줄 관리의 어려움



반려견을 맡기거나 픽업하러 가야하는
반려인의 입장에서 여러가지 이유로
스케줄을 관리하기 어려울 수 있습니다.

2

서비스에 대한 믿음 부족



일일이 여러곳의 리뷰를 비교해야하며,
문제 행동 혹은 질환이 있는 반려견을
키우는 경우, 더욱 신경써야합니다.

3

이동 부담 증가



근처에 샵이 없어 멀리 이동하면,
교통수단에서 반려인에게 부담을 주며,
반려견의 스트레스에도 영향을 줍니다.



TARGET USER

바쁜 일상 속
여유가 없는

1인 가구

각각의 특징에 맞는
맞춤 케어가 필요한

다견 가정

무게와 크기로 인해
혼자 케어하기 힘든

중, 대형견 견주

스트레스를 최소화 해줄
전문가의 손길이 필요한

노령견 견주

“ 집 앞까지 찾아가는 1:1 맞춤 케어 서비스 ”

BRAND CONCEPT

깔끔하고 직관적인 인터페이스로 사용자들에게 편리한 경험을 제공하며,
블루톤을 사용하여 신뢰감과 더불어 세련된 브랜드 이미지를 구축합니다.

NAVY

#001A63

NAVY

#001A63

NAVY

#001A63



PAW는 강아지를 의미하고, 강아지의 상태에 맞춰 서비스한다는
의미의 FIT을 붙여 브랜드의 의도를 그대로 담은 이름

프리텐다드 PRETENDARD

맞춤 서비스로 즐거운 경험을 제공하는, PAWFIT

01
OVERVIEW

02

ARCHITECTURE

프로젝트 구조 및 역할 분담



03
DATABASE

04
CODE

05
DEMONSTRATION

USE CASE - USER



USE CASE - ADMIN



TEAM PAWFIT



김소미 팀장

- 프로젝트 총괄, 형상 관리
- ERD 설계
- 와이어프레임 및 디자인
- 브랜드 소개, 고객센터 반응형 웹 제작 및 JS / React 활용 정적, 동적 UI, 고객센터 페이지 기능 개발
- 프로젝트 발표 자료 제작



하지현 기술리더

- 원격 DB 환경 구축
- 관리자 페이지 기능 개발
- Chart.js / D3.js 활용 그래프 제작
- 뉴스레터 제작 및 전송 구현
- 구글 클라우드 이용 프로젝트 배포



김민진 FRONT

- DB 데이터 생성
- 서비스 / 마이페이지 반응형 웹 제작 및 JS / React 활용 정적, 동적 UI
- PWA 활용 웹앱 설정
- 프로젝트 발표



박진수 FRONT

- 프론트엔드 패키징, 라우팅 설정
- 로그인 / 관리자 페이지 반응형 웹 제작 및 JS / React 활용 정적, 동적 UI
- 다음 우편번호 API
- 랜딩페이지 제작

TEAM PAWFIT



김은하 BACK

- 로그인 / 회원가입 페이지 기능 개발
- Java Mail 활용 이메일 전송 구현
- Spring Security 활용 비밀번호 암호화
- ID / PW 찾기 기능 개발



임동규 BACK

- ERD 설계
- 초기 개발 환경 구축 및 백엔드 패키징
- 차량 추천 및 목록 / 예약 기능 개발
- 프로젝트 발표



정주호 BACK

- 마이페이지 기능 개발
- Python 활용 견종 분석 기능 개발
- AWS 활용 이미지 저장
- 소셜로그인 기능 구현 (카카오/네이버)

01

OVERVIEW

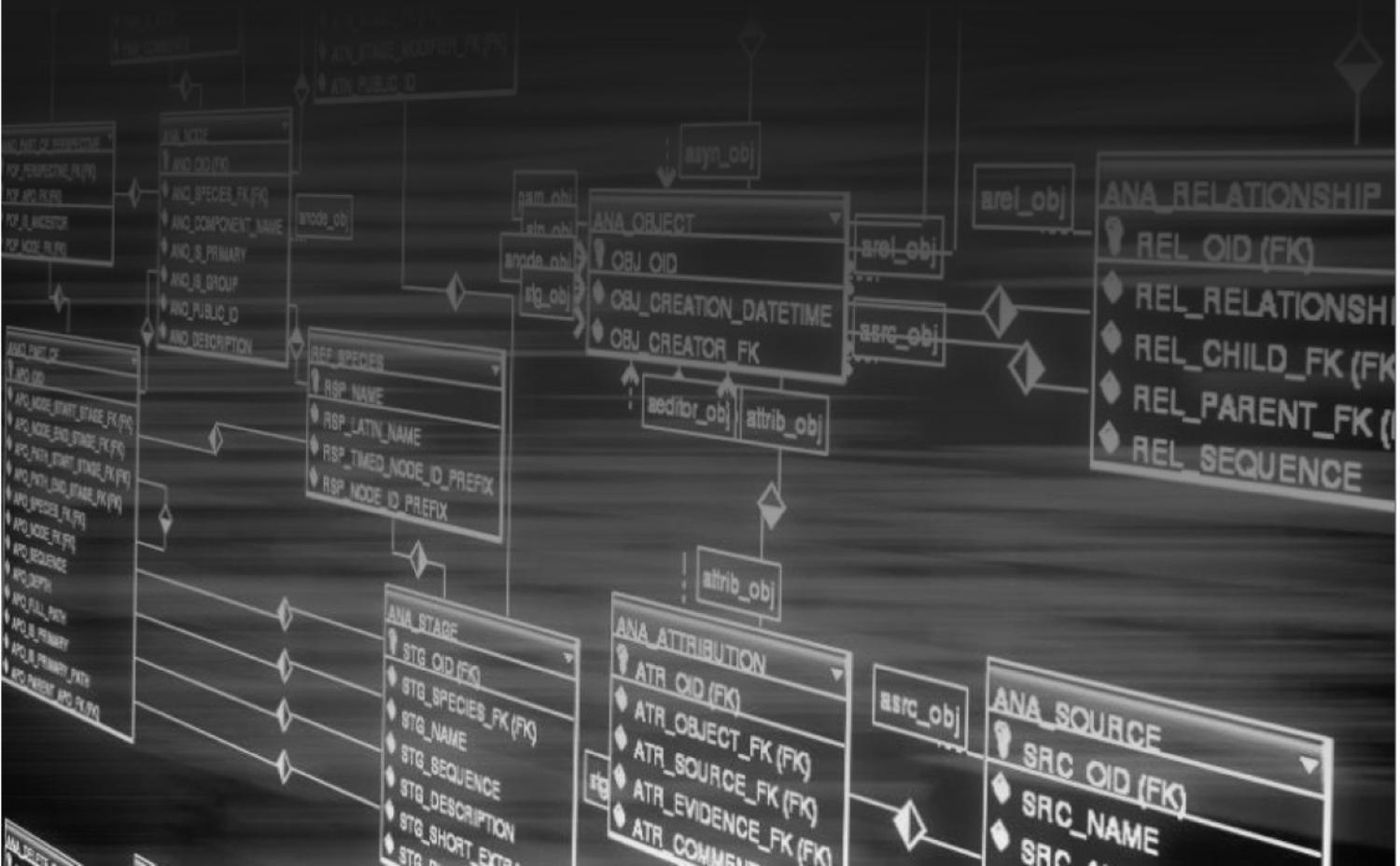
02

ARCHITECTURE

03

DATABASE

개발 환경 및 ERD 설계



04

CODE

05

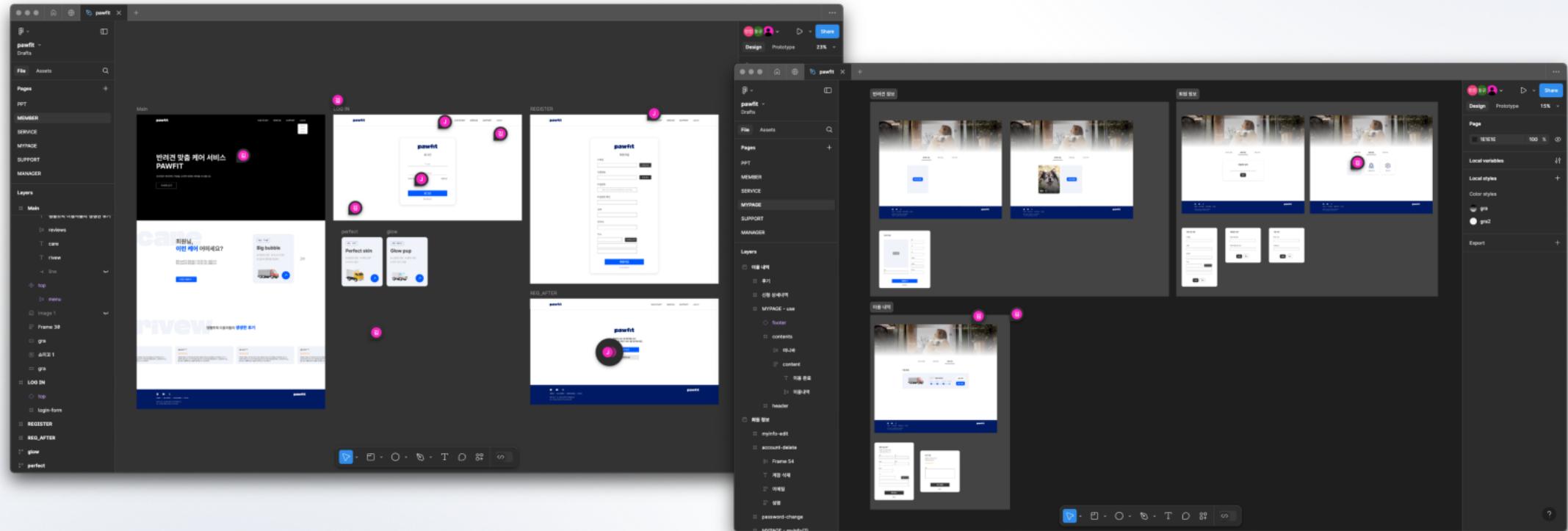
DEMONSTRATION

ENVIRONMENT



개발 환경		언어 및 프레임 워크		기타 툴	
OS	Mac OS, Windows	언어 / 기술	JAVA, Java Script, HTML, CSS, SQL, Python	커뮤니케이션	Slack, Notion
Web Server	Apache Tomcat, Google Cloud, AWS S3	프레임워크	SpringBoot	형상 관리	Git, GitHub
Data Base	Maria DB	라이브러리	React, My Batis, Chart.js, D3.js, JWT, Spring Security, Java Mail, Keras, Tensorflow	오픈 API	Kakao Login, Naver Login, 다음 우편번호
Build	Maven			디자인	Figma, Illustration, Photo Shop
IDE	IntelliJ, VS Code				

DESIGN



FIGMA / ILLUSTRATION / PHOTOSHOP

- 프로젝트의 시각적 퀄리티 향상을 위한 이미지 보정 및 제작
- 오토 레이아웃을 활용한 인터페이스 제작으로 html, css 작성 용이
- 전체적인 레이아웃을 통한 웹의 시각적 작동 구조 파악 가능

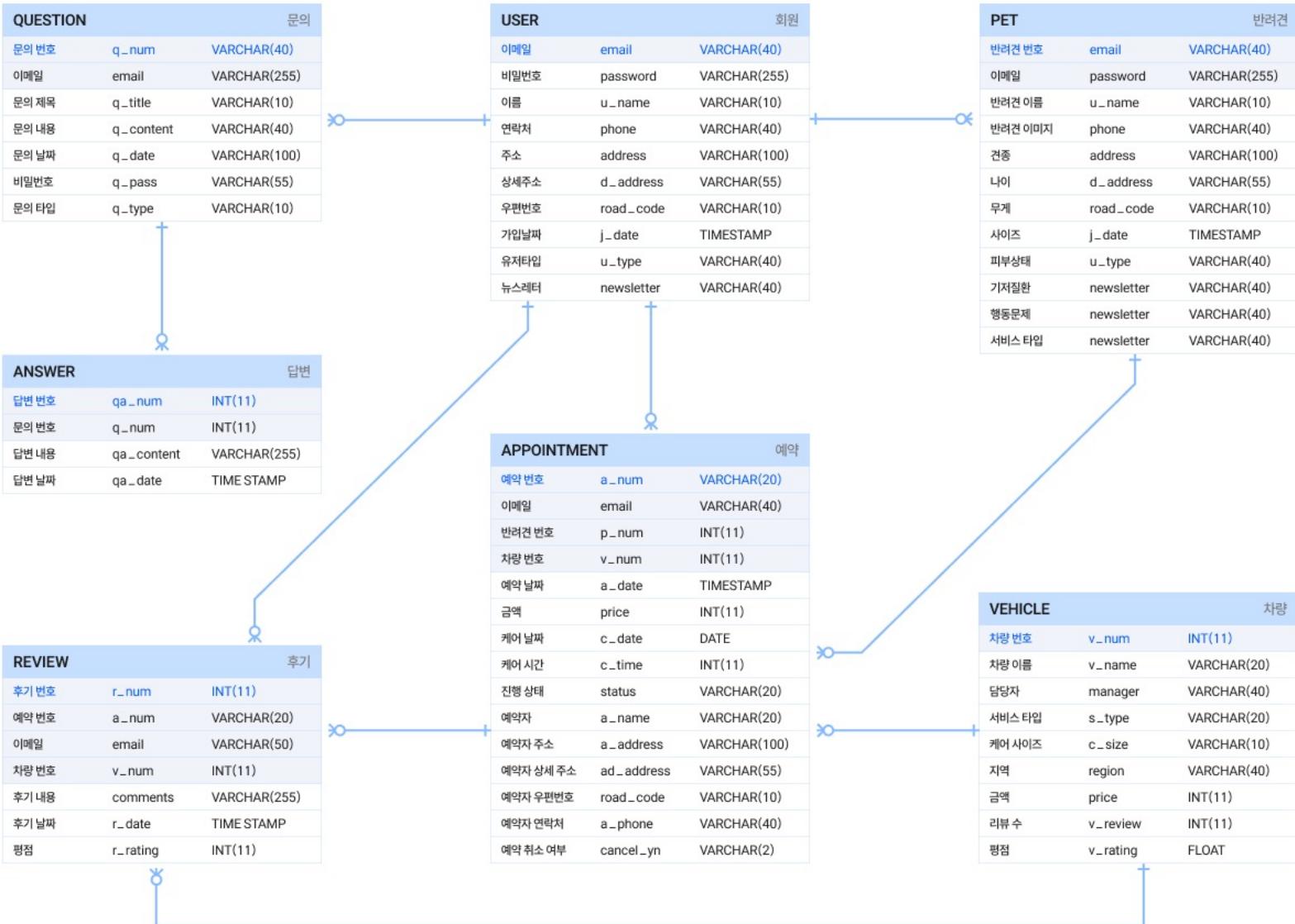
ERD



- PRIMARY KEY**
- FOREIGN KEY**
- ATTRIBUTE**

DELETED_USER			탈퇴회원
탈퇴 번호	id	INT(11)	
탈퇴 날짜	deleted_at	TIME STAMP	

IMAGES			이미지
이미지 번호	id	BIGINT(20)	
파일명	file_name	VARCHAR(255)	
파일 경로	s3_url	VARCHAR(500)	



01

OVERVIEW

02

ARCHITECTURE

03

DATABASE

04

CODE

기능별 주요 코드



05

DEMONSTRATION

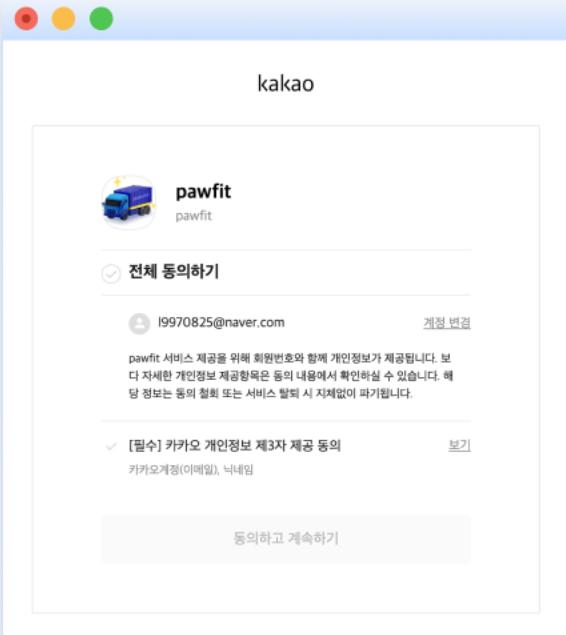
MEMBER

로그인 / 회원가입

A screenshot of a Chrome browser window showing the Pawfit login page. The URL in the address bar is `pawfit.duckdns.org/login`. The page features a large central input field labeled "로그인" (Login). Below it are fields for "E-mail" and "Password". At the bottom of the input area are two social login buttons: "카카오 로그인" (Kakao Login) and "네이버 로그인" (Naver Login). A blue "로그인" button is located at the bottom right of the input field. Navigation links "이메일 찾기 | 비밀번호 찾기" and "회원 가입" are positioned at the bottom left and right respectively. The top navigation bar includes links for "BRAND", "SERVICE", "SUPPORT", and "LOG IN". The browser's toolbar and status bar are visible at the top.

A screenshot of a mobile phone displaying the Pawfit member sign-up page. The screen shows a form with fields for "이메일" (Email) containing "example@pawfit.com", a "인증번호" (Verification Number) field with a "인증" (Verify) button, and a "비밀번호" (Password) field with a "확인" (Check) button. Below these are fields for "비밀번호 확인" (Confirm Password), "이름" (Name), and "연락처" (Contact). The top navigation bar includes a logo and a three-dot menu icon. The time "06:00" and battery level "100" are shown at the top of the phone screen.

소셜 로그인



```
@GetMapping("/callback")
public void kakaoLogin(@RequestParam String code, HttpServletResponse response) {
    try {
        // 1. 카카오에서 Access Token 받기 (한 번만 유효)
        String accessToken = getKakaoAccessToken(code);

        // 2. Access Token으로 사용자 정보 가져오기
        Map<String, String> kakaoUserInfo = getKakaoUserInfo(accessToken);
        String email = kakaoUserInfo.get("email");

        // 3. 이메일로 사용자 정보 조회
        UserAuthDto user = userAuthMapper.findByEmail(email);

        String jwtToken;
        ...
        // JWT 토큰을 쿠키에 저장
        addJwtToCookie(response, jwtToken);

        response.sendRedirect("https://pawfit.duckdns.org");
    }
    ...
}
```

```
// Access Token을 얻는 메서드
private String getKakaoAccessToken(String code) {
    try {
        RestTemplate restTemplate = new RestTemplate();

        String url = "https://kauth.kakao.com/oauth/token";
        UriComponentsBuilder builder = UriComponentsBuilder.fromHttpUrl(url);

        // 요청 본문 데이터 생성
        MultiValueMap<String, String> body = new LinkedMultiValueMap<>();
        body.add("grant_type", "authorization_code");
        body.add("client_id", kakaoClientId);
        body.add("redirect_uri", kakaoRedirectUri);
        body.add("code", code);

        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);

        // 요청 본문과 헤더를 포함한 HttpEntity 생성
        HttpEntity<MultiValueMap<String, String>> entity = new HttpEntity<>(body, headers);
        ResponseEntity<Map> response = restTemplate.postForEntity(builder.toUriString(), entity, Map.class);

        System.out.println("카카오 Access Token 응답 상태: " + response.getStatusCode());
        ...
    }
}
```



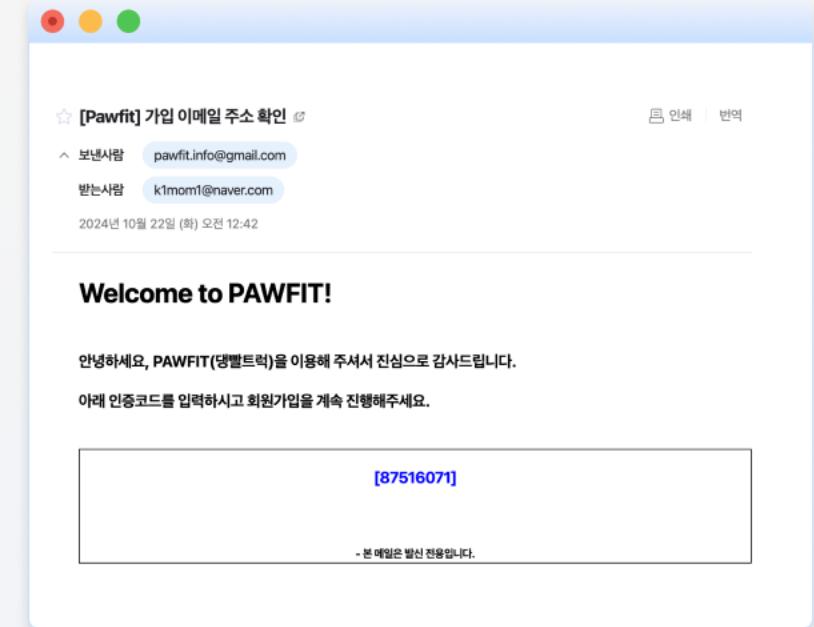
KAKAO / NAVER

- OAUTH2 인증을 통해 사용자의 정보를 가져오고, JWT를 생성하여 사용자 인증을 수행

회원가입 이메일 인증

```
// 이메일 품 객체 생성
public void createEmailForm(String setFrom, String toEmail, String title, String contents) {
    MimeMessage message = javaMailSender.createMimeMessage();

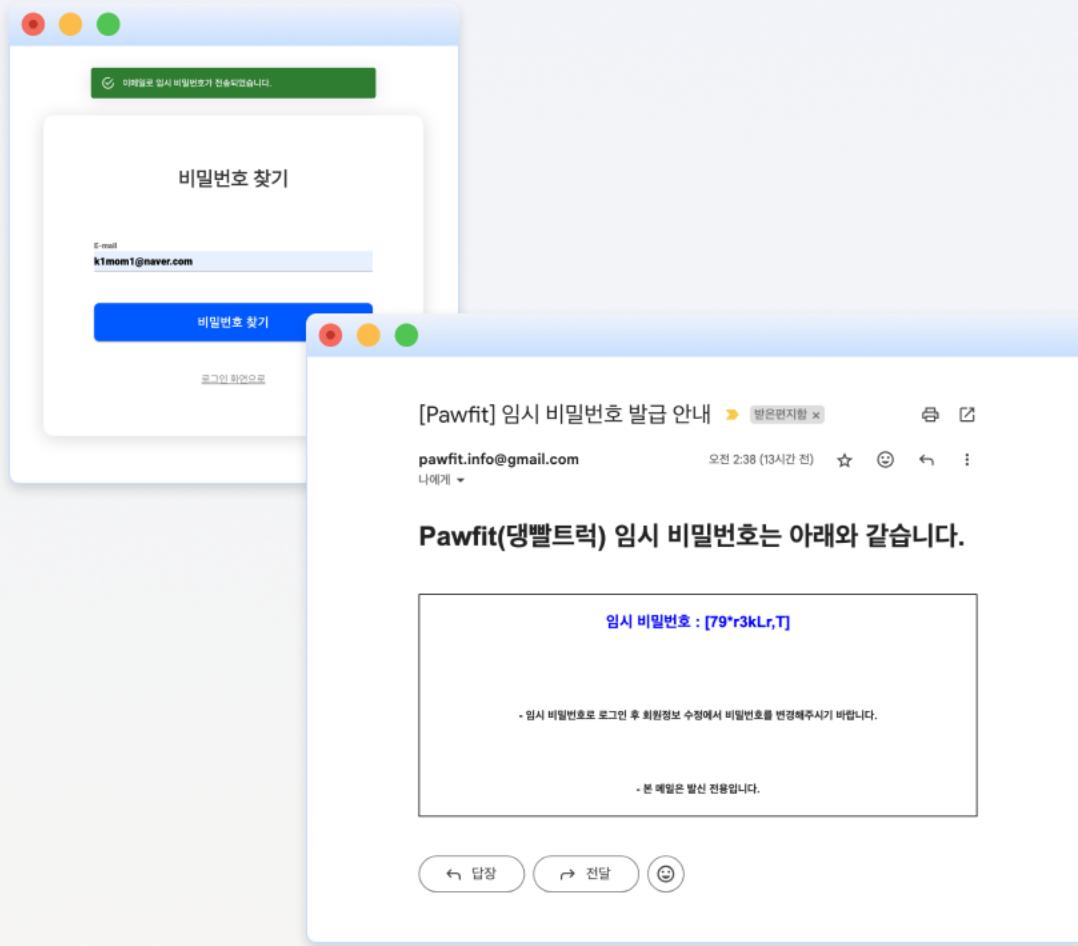
    try {
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "utf-8");
        messageHelper.setFrom(setFrom);
        messageHelper.setTo(toEmail);
        messageHelper.setSubject(title);
        messageHelper.setText(contents, true);
        javaMailSender.send(message);
    } catch (MessagingException e) {
        e.printStackTrace();
        log.error("Failed to send email: " + e.getMessage(), e);
    }
}
```



JAVA MAIL

- 회원가입 시 사용자가 입력한 이메일로 랜덤으로 생성된 인증번호를 전송
- 비밀번호 분실 시 사용자가 입력한 이메일이 DB에 있는지 확인 후 임시 비밀번호 발급

비밀번호 찾기



```
// 이메일 발송 및 비밀번호 데이터 저장(임시 비밀번호 인증)
public String sendTempPassword(EmailDto emailDto) throws Exception {
    createTempPassword(10);

    saveTempPassword(emailDto);
    System.out.println(emailDto.getPassword());

    String setFrom = "pawfit@gmail.com";
    String toEmail = emailDto.getEmail();
    String title = "[Pawfit] 임시 비밀번호 발급 안내";

    String contents = "";
    contents += "<h1>Pawfit(댕빨트럭) 임시 비밀번호는 아래와 같습니다.</h1>";
    contents += "<br>";
    contents += "<div align=\"center\" style=\"border:1px solid black;\">";
    contents += "<h3 style=\"color:blue\">임시 비밀번호 : <strong>[" +
        + tempPassword + "]</strong></h3><br>";
    contents += "<br><h5>- 임시 비밀번호로 로그인 후 회원정보 수정에서 비밀번호를 변경해주시기 바랍니다.</h5>";
    contents += "<br><h5>- 본 메일은 발신 전용입니다.</h5>";
    contents += "</div>";

    createEmailForm(setFrom, toEmail, title, contents);

    return "임시 비밀번호 발송 완료 " + tempPassword;
}
```

비밀번호 암호화

admin4	\$2a\$10\$fuB560CzXTcP8bxrW5Zk/ObNDRu06YiFwfOqrR170xp0Rxj4ZN7W8W
김은하	\$2a\$10\$XQkAxup71YGcVFxtYol2yuVrkqxDEm0171RIQ3dJzDabqvpHYKQgi
김민진	\$2a\$10\$3pDX/0yfsQuBalqZmUvMmegjcxAsDKR2Ypzkf6eL5eEl3eqJ/Us0S
김민진	\$2a\$10\$prZlnKe.zHfmoQlup6X7TeYqrxFYpDxnHjdK7WJNB4rOZk/T1HQ2.
임동규	\$2a\$10\$BaaUdVO9t96Rn0SuQEjZPeEV0/8BsElJC1dmNaGweR5Eri0MrMBVy
박진수	\$2a\$10\$mKps4DnovxqkdQQaeweMqe6ulhn.pGj8b0yW7SjTy4TY77DeGtI3e
하지현	\$2a\$10\$/2cksmcWXsrcs3CKsxxLXOU3vHNCfO7PockgF54WV2MLAXh2D3lps
김지원	\$2a\$10\$cIAfY64afLfiP4krW1gBOubTO3tPDmfAYqpkeBUq3F/lngNVhEVIK
정옥순	\$2a\$10\$TP3JiOwp0m8cdH./dA3P7uSz5VHz7ogR6VYgQurgt/NQhu2bzPzo6
김소미	\$2a\$10\$FnCXl/7yWGvGSLcENpRc.ezc8/wVbdFWuxebiWBLPJ/yXknJ5J4zK

```
private final RegisterMapper registerMapper;
private final BCryptPasswordEncoder passwordEncoder;
private static final HashMap<String, String> matchPassword = new HashMap<>();
private static final String PASSWORD_EXPRESSION =
"^(?=.*[a-zA-Z])(?=.*[\\d])(?=.*[@#$%^&*()_+|~\\-=`{}\\[\\]\\:\\\"';'<>?,./]).{8,15}$";
```

// 회원가입 정보 user 테이블에 넣기(user)

```
public String addUser(RegisterDto registerDto) {
    System.out.println("userName: " + registerDto.getUserName());

    // 사용자 생성
    RegisterDto newUser = new RegisterDto();

    newUser.setEmail(registerDto.getEmail());
    newUser.setPassword(passwordEncoder.encode(registerDto.getPassword()));
    newUser.setUserName(registerDto.getUserName());
    newUser.setPhone(registerDto.getPhone());
    newUser.setAddress(registerDto.getAddress());
    newUser.setNewsletter(registerDto.getNewsletter());
    newUser.setDetailAddress(registerDto.getDetailAddress());
    newUser.setRoadCode(registerDto.getRoadCode());

    registerMapper.addUser(newUser);

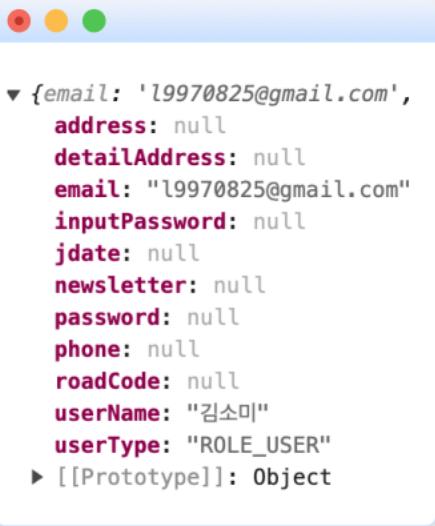
    return "정상적으로 가입이 완료되었습니다.";
}
```



SPRING SECURITY

- 사용자가 입력한 비밀번호를 암호화하여 인코딩한 변수를 DB에 저장하여 사용자 정보 보안 강화

사용자 토큰



```
// 토큰 생성
public String createJwt(String email, String role, long expiredMs) {
    System.out.println("email : " + email);
    System.out.println("role : " + role);
    String token = Jwts.builder()
        .claim("email", email)
        .claim("role", role)
        .issuedAt(new Date(System.currentTimeMillis()))
        .expiration(new Date(System.currentTimeMillis() + expiredMs))
        .signWith(secretKey)
        .compact();

    System.out.println("Generated Token: " + token);

    return token;
}
```

```
export const loadUser = () => (dispatch) => {
  let token = getCookie("Authorization"); // 쿠키에서 토큰 가져오기

  if (!token) {
    console.error("Token not found in cookies");
  }

  if (token && token.split(".").length === 3) {
    const decoded = jwtDecode(token);
    const expirationTime = decoded.exp * 1000;
    const currentTime = Date.now();

    if (expirationTime < currentTime) {
      dispatch(logout());
    } else {
      axios.defaults.headers.common["Authorization"] = `Bearer ${token}`;
      dispatch(loginSuccess({ token }));

      // 만료 시간 전까지 자동 로그아웃 설정
      if (logoutTimer) clearTimeout(logoutTimer);
      logoutTimer = setTimeout(() => {
        dispatch(logout());
      }, expirationTime - currentTime);
    }
  } else {
    console.error("Invalid token format or token not found:", token);
    dispatch(logout());
  };
};
```



JWT

- 백엔드에서 EMAIL, ROLE, ISSUEDAT(현재 토큰 발행 시간), EXPIRATION(토큰 만료 시간)을 넣어주어 토큰 생성
- 프론트에서 요청이 들어올 때마다 JWT 토큰이 포함되어있는지 확인하고 유효한지 검증

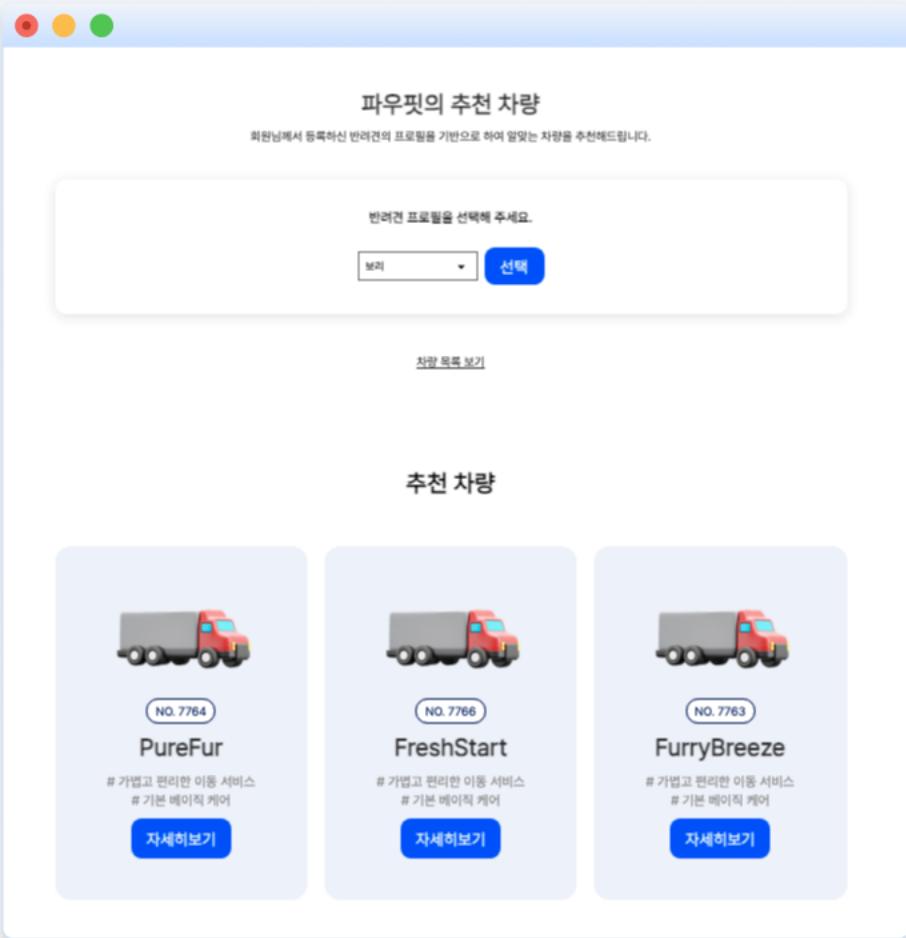
SERVICE

차량추천 / 서비스 예약

The screenshot shows a desktop browser window displaying the Pawfit Service website. The URL in the address bar is pawfit.duckdns.org/service. The page features a large, slightly blurred background image of a wet, brown dog looking up. Overlaid on this image is the word "SERVICE". At the top left is the Pawfit logo, and at the top right are navigation links: BRAND, SERVICE, SUPPORT, and 김소미님 (Kimsomin). Below the main image, there's a section titled "파우핏의 추천 차량" (Pawfit Recommended Vehicles) with a note: "회원님께서 등록하신 반려견의 프로필을 기반으로 하여 알맞는 차량을 추천해드립니다." (Based on your registered pet's profile, we will recommend appropriate vehicles). A dropdown menu allows users to select a vehicle type, with "보리" (Bori) currently selected. A blue "선택" (Select) button is positioned next to the dropdown. At the bottom right of the page is a blue "예약하기" (Book Now) button.

The screenshot shows a smartphone displaying the Pawfit Service app. The screen has a dark header with the Pawfit logo and a three-line menu icon. Below the header is a thumbnail image of a small, white fluffy dog being groomed. To the right of the thumbnail, the text "NO. 7761" is displayed in a small oval. Underneath the image, the service name "GentleBath" is shown in bold. Below "GentleBath" are hashtags: "#중·대형견전문 #경력10년차" and "#파우핏대표차량". There is also a rating section showing five empty star icons with "(0)" next to them. Further down, there are sections for "Price" (with options "일반 / 소형견" and "100,000원") and a blue "예약하기" (Book Now) button at the bottom right.

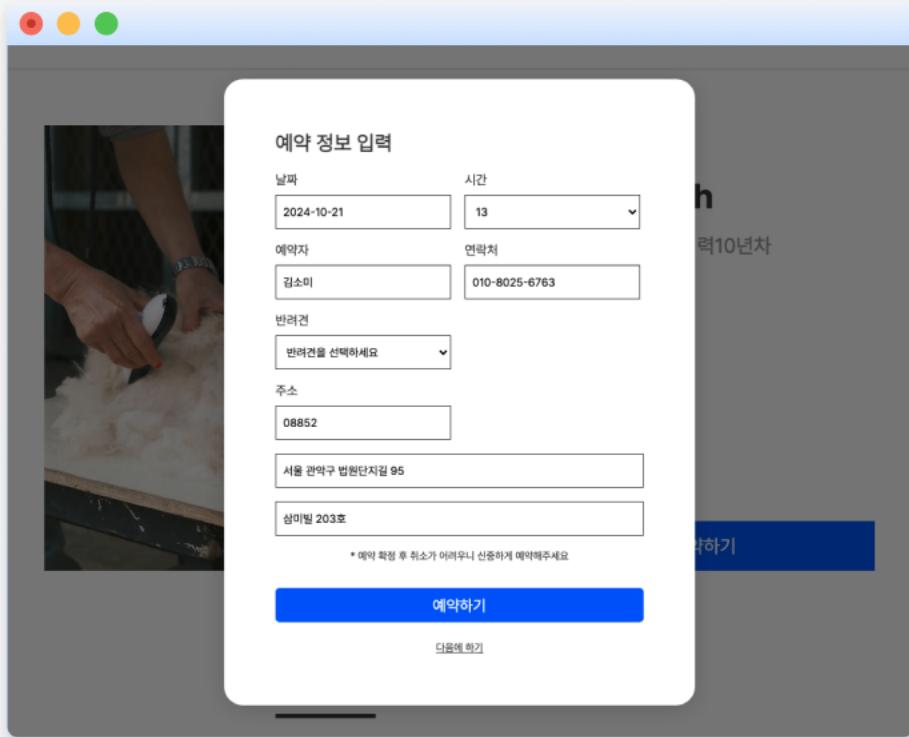
차량 추천



```
@GetMapping("/recommend")
public ResponseEntity<Object> recommends(
    // 비회원도 조회 할수 있도록 required = false로 설정
    @RequestParam(required = false) String email) {
    // 이메일을 파라미터로 받아 해당이메일에 반려견이 몇마리 등록되어 있는지 확인
    int petCount = recommendService.getPetCount(email);

    if(petCount == 0) { // 등록된 강아지 정보가 없을때
        // 강아지 정보가 없을때 실행되는 메서드
        List<Recommend> recommend =
            recommendService.recommendNoPet(email);
        return ResponseEntity.ok(recommend);
    }else if (petCount == 1) { // 강아지가 한마리 등록되어 있을때
        // 해당 email에 등록되어 있는 강아지 정보와 함께 추천 차량 조회
        List<Recommend> recommend =
            recommendService.recommendOnePet(email);
        return ResponseEntity.ok(recommend);
    }else{ // 강아지가 여러마리일 경우, 차량 추천을 위해
        // 등록되어 있는 강아지 리스트 전달
        List<Recommend> recommend =
            recommendService.getPetList(email);
        return ResponseEntity.ok(recommend);
    }
}
```

서비스 예약



```
CREATE DEFINER='root'@'%' TRIGGER before_insert_appointment  
BEFORE INSERT ON appointment
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE formatted_date VARCHAR(20);
```

```
-- 예약 날짜를 기반으로 포맷팅 (yyMMddHHmmss)
```

```
SET formatted_date = DATE_FORMAT(NEW.a_date, '%Y%m%d%H%i%s');
```

```
-- 예약 번호 생성
```

```
SET NEW.a_num = formatted_date;
```

```
END
```

```
CREATE EVENT update_status_at_midnight  
ON SCHEDULE EVERY 1 DAY
```

```
STARTS '2024-09-24 02:00:00.000'
```

```
ON COMPLETION NOT PRESERVE
```

```
ENABLE
```

```
DO BEGIN
```

```
UPDATE appointment
```

```
SET status = '진행'
```

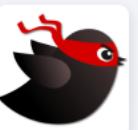
```
WHERE DATE(c_date) = CURDATE() AND status != '진행';
```

```
UPDATE appointment
```

```
SET status = '완료'
```

```
WHERE DATE (c_date) < CURDATE() AND status != '완료';
```

```
END
```



MY BATIS / MARIA DB

- 사용자가 회원가입 시 입력한 정보를 불러와 수정 가능하고, 반려견 및 예약 날짜를 지정하여 예약 가능
- 서비스 예약 시 예약번호 자동 생성 트리거 작성, 신청 날짜가 지남에 따라 진행/완료로 상태가 바뀌는 이벤트 스케줄러 작성

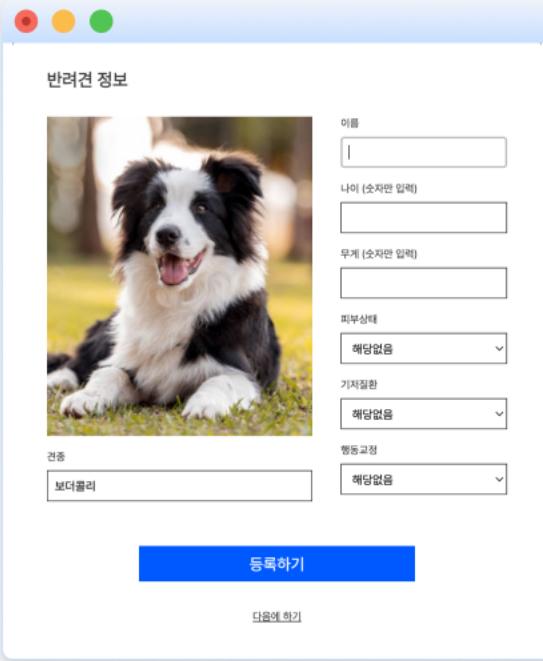
MY PAGE

반려견 정보 / 회원 정보 / 이용 내역

The screenshot shows the Pawfit My Page interface on a desktop browser. The header features a large blue circular logo and the text "MY PAGE". Below the header, there are tabs for "반려견 정보", "회원 정보", and "이용 내역". The "반려견 정보" tab is currently selected. A large image of a woman holding a dog is displayed prominently. At the bottom left, there is a photo of a dog with a "반려견 등록" button. The browser's address bar shows the URL: pawfit.duckdns.org/mypage/pet-info.

The screenshot shows the Pawfit My Page interface on a mobile phone. The top navigation bar includes the Pawfit logo, a menu icon, and the time "06:00". The main content area displays the same "MY PAGE" header and tabs as the desktop version. Below the tabs, a section titled "케어신청(2024-10-21)" shows a cartoon illustration of a blue delivery truck with the text "BigBubble" and "NO. 7787". The bottom of the screen shows a progress bar with several blue dots.

반려견 정보 - 견종 분류



```
# 데이터 증강 설정
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2 # 검증 데이터 비율 설정
)

# 모델 구축
model = Sequential([
    Input(shape=(64, 64, 3)),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])

# 모델 컴파일
model.compile(loss='categorical_crossentropy',
              optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
              metrics=['accuracy'])

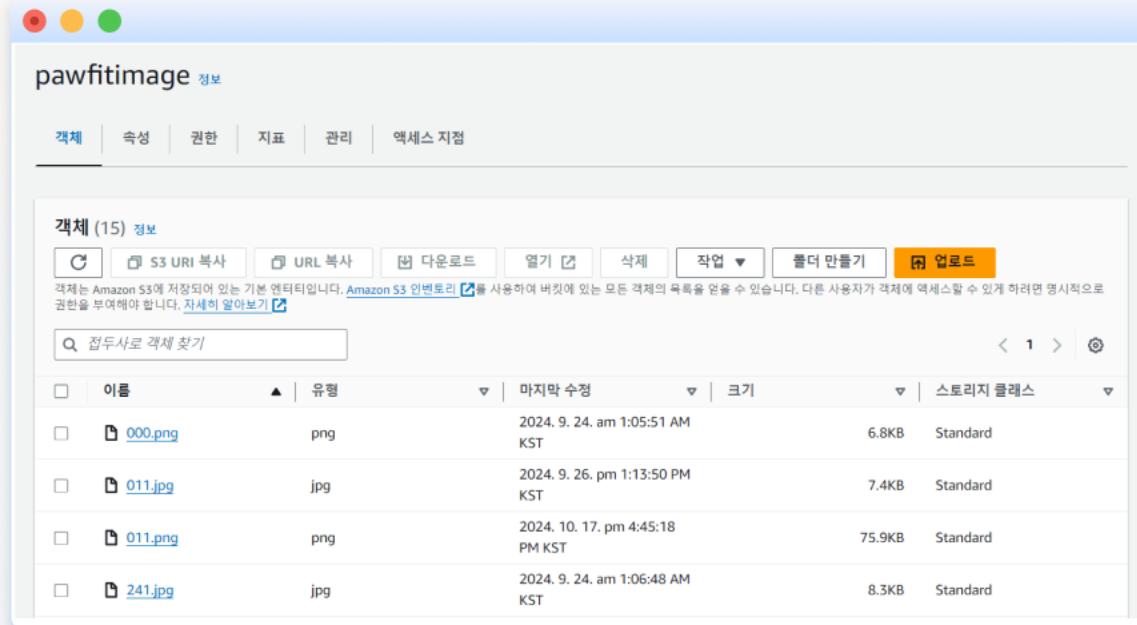
# 조기 종료 및 체크포인트 설정
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
model_checkpoint = ModelCheckpoint(model_path, save_best_only=True, monitor='val_loss')
```



PYTHON / KERAS / TENSORFLOW

- 사용자가 업로드한 이미지를 받아서 견종을 예측
- CONV2D 레이어를 추가하고 필터 수를 증가시켜 모델 아키텍처 개선
- EARLYSTOPPING과 MODELCHECKPOINT를 사용하여 최적의 모델을 유지

반려견 정보 - 이미지 저장



The screenshot shows the AWS S3 console for the 'pawfitimage' bucket. It displays a list of 15 objects, all being image files (png or jpg) with sizes ranging from 6.8KB to 8.3KB. The files were uploaded between September 24 and October 17, 2024. The interface includes standard S3 management tools like copy, delete, and download.

이름	유형	마지막 수정	크기	스토리지 클래스
000.png	png	2024. 9. 24. am 1:05:51 AM KST	6.8KB	Standard
011.jpg	jpg	2024. 9. 26. pm 1:13:50 PM KST	7.4KB	Standard
011.png	png	2024. 10. 17. pm 4:45:18 PM KST	75.9KB	Standard
241.jpg	jpg	2024. 9. 24. am 1:06:48 AM KST	8.3KB	Standard

```
public String uploadFile(MultipartFile file) {
    String fileName = file.getOriginalFilename(); // 파일 이름 가져오기

    try (InputStream inputStream = file.getInputStream()) {
        PutObjectRequest request = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(fileName)
            .build();

        s3Client.putObject(request, RequestBody.fromInputStream(inputStream, file.getSize()));
        logger.info("파일이 S3에 성공적으로 업로드되었습니다: {}", fileName);
        return String.format("https://%.s.s3.amazonaws.com/%s", bucketName, fileName);
    } catch (S3Exception e) {
        logger.error("S3 파일 업로드 실패: {}", e.awsErrorDetails().errorMessage());
        throw new RuntimeException("S3 파일 업로드 실패: " + e.awsErrorDetails().errorMessage());
    } catch (IOException e) {
        logger.error("파일 스트림 처리 실패: {}", e.getMessage());
        throw new RuntimeException("파일 스트림 처리 실패: " + e.getMessage());
    }
}
```



AWS S3

- AWS S3를 사용하여 이미지 파일을 업로드하고 삭제하는 기능을 구현

SUPPORT

자주 묻는 질문 / 문의 게시판

Chrome 파일 설정 보기 방문 기록 북마크 프로필 덤 창 도움말

PAWFIT pawfit.duckdns.org/support

pawfit BRAND SERVICE SUPPORT 김소님

자주 묻는 질문 문의 게시판

무엇이 궁금하신가요?

고객센터 1588-5757

평일 : 09:30~17:00(12:00~13:00 점심시간)
주말 : 09:30~17:00

06:00

06:00

10월 21일 (화) 09:31

100%

pawfit

자주 묻는 질문 문의 게시판

총 9개의 문의가 있습니다.

작성

[서비스] 서비스 문의입니다.	김소*	답변대기
[예약] 예약날짜 변경 문의	김소*	답변완료
[서비스] 사용 제품	김소*	답변완료
[예약] 문의	임동*	답변완료
[기타] 담당자 문의드립니다.	admin*	답변완료
[서비스] 옷시간 조율하나요??	김정*	답변완료

문의게시판

The screenshot shows a web application interface for a question and answer board. At the top, there are three colored circular icons (red, yellow, green). Below them, a message says "총 10개의 문의가 있습니다." (There are 10 inquiries). A blue button labeled "▶ 목록" (List) is visible.

The main area displays a table with columns: No., 제목 (Title), 작성자 (Author), 날짜 (Date), and 상태 (Status). Two entries are shown:

No.	제목	작성자	날짜	상태
10	[서비스] 서비스 문의입니다.	김소*	2024-10-17	답변대기
9	[예약] 예약날짜 변경 문의	김소*	2024-10-16	답변완료

Below the table, a list of inquiries is shown, each with a red, yellow, and green circular icon. The items are:

- 7 [예약] 문의
- 6 [기타] 담당자 문의드립니다.
- 5 [서비스] 몇시간 소요되나요??
- 4 [예약] 한번에 두마리 예약
- 3 [기타] 문의드립니다. ◀
- 2 [회원] 회원정보 변경문의 ◀
- 1 [예약] 예약 확정되었는데 정보 변경하고 싶어요. ◀

A vertical page number "1" is at the bottom left. A blue button labeled "▶ 목록" is at the bottom right.

In the center, a detailed view of inquiry No. 2 is shown. The title is "회원정보 변경문의". It was posted by "작성자 김소미" on "작성날짜 2024-09-26". The content asks if it's possible to change the reservation date. A response from "운영자 답변" dated "2024-09-26" says: "이메일을 변경하시려면 탈퇴 후 재가입하셔야합니다!" (You must withdraw and re-register if you want to change your email address).



MY BATIS

- MY BATIS로 문의글 관련 CRUD 작성
- 로그인 시 문의글 열람 및 작성이 가능하며, 관리자일 경우 답변 작성 가능

```
<!-- 문의사항 전체보기 -->
<select id="getAllQuestion" resultType="com.backend.dto.question.Question">
    SELECT
        q.q_num AS qNum,
        q.q_title AS qTitle,
        q.q_content AS qContent,
        DATE_FORMAT(q.q_date, '%Y-%m-%d') AS qDate,
        q.email AS email,
        q.q_type AS qType,
        q.q_pass AS qPass,
        a.qa_content AS qaContent,
        CASE
            WHEN LENGTH(u.u_name) > 0 THEN
                CONCAT(SUBSTRING(u.u_name, 1, CHAR_LENGTH(u.u_name) - 1), '*')
            ELSE
                u.u_name
            END AS uName
    FROM
        question q
        JOIN user u ON q.email = u.email
        LEFT JOIN answer a ON q.q_num = a.q_num
    ORDER BY q.q_num desc
    LIMIT #{limit} OFFSET #{offset}
</select>

<!-- 문의사항 게시글 자세히보기 -->
<select id="viewQuestion" parameterType="int" resultType="com.backend.dto.question.Question">
    SELECT
        q.q_num AS qNum,
        q.q_title AS qTitle,
        q.q_content AS qContent,
        q.q_type AS qType,
        DATE_FORMAT(q.q_date, '%Y-%m-%d') AS qDate,
        q.email AS email, -- 이메일을 가져오되, User 객체의 필드로 매핑
        u.u_name AS uName,
        a.qa_content AS qaContent, -- answer 테이블의 qa_content를 가져옴
        DATE_FORMAT(a.qa_date, '%Y-%m-%d') AS qaDate -- 불필요한 썸표 제거
    FROM question q
        JOIN user u ON q.email = u.email
        LEFT JOIN answer a ON q.q_num = a.q_num -- answer 테이블과 조인
    WHERE q.q_num = #{qNum}
</select>
```

MANAGER

회원 / 예약 / 차량 / 후기 관리

Chrome 파일 수정 보기 방문 기록 북마크 프로필 백 창 도움말

pawfit.pawfit.duckdns.org/manager/board 10월 21일 (월) 09:36

홈페이지 바로가기

보드 회원 예약 후기 차량

2024-10-21 월

관리자님, 안녕하세요! 실시간 PAWFIT 데이터를 보여드릴게요.

금일 가입자 수 0 명 금일 신청 수 1 명 금일 후기 수 0 건 미답변 문의 1 건

09월 5주차 10월 1주차 10월 2주차 10월 3주차

가입자 수
탈퇴자 수

09월 5주차 10월 1주차 10월 2주차 10월 3주차

서비스 신청 수
서비스 취소 수

admin4 관리자님

주기	가입자 수	탈퇴자 수
09월 5주차	3	1
10월 1주차	6	5
10월 2주차	3	1
10월 3주차	10	4

주기	신청 수	취소 수
09월 5주차	2	0
10월 1주차	7	4
10월 2주차	6	3
10월 3주차	4	1

관리자 접근 권한



```
// csrf 비활성화 및 필터 설정
http
    .csrf(csrf -> csrf.disable()) // CSRF 비활성화
    .formLogin(form -> form.disable()) // Form 로그인 비활성화
    .httpBasic(basic -> basic.disable()) // HTTP Basic 인증 비활성화
    .authorizeHttpRequests(auth -> auth
        .requestMatchers("/admin/**").hasRole("ADMIN") // 관리자 접근 제어
        .anyRequest().permitAll() // 나머지 요청은 모두 허용
    .addFilterAt(loginFilter, UsernamePasswordAuthenticationFilter.class) // 커스텀 로그인 필터 등록
    .addFilterAfter(new JWTFilter(jwtUtil), UsernamePasswordAuthenticationFilter.class) // JWT 필터 등록
    .sessionManagement(session -> session
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS)) // 세션 사용 안함
    .exceptionHandling(exception -> exception
        .authenticationEntryPoint(jwtAuthenticationEntryPoint) // 401 예외 처리
        .accessDeniedHandler(jwtAccessDeniedHandler)) // 403 예외 처리
    .cors(withDefaults()); // CORS 설정 추가
```

```
/* 관리자 페이지 라우트 */
<Route
path="/manager"
element={
    userType === "ROLE_ADMIN" ? (
        <ManagerMain />
    ) : (
        <UnauthorizedAccess />
    )
}>
```



JWT / SPRING SECURITY

- 관리자 페이지에 일반 유저가 접근할 수 없도록 백/프론트 모두에서 제어 하여 보안 강화

메인 보드 - 시각화 그래프



D3.JS

- D3.JS 활용 주간 가입/탈퇴자 수, 서비스 신청/취소 수를 나타내는 그래프 구현
- 해당 시각화를 통해 고객들의 동향을 파악하고, 마케팅 계획을 세우기에 유용

```
// 데이터 정렬
const sortedData = response.data.sort((a, b) => {
  const [yearA, monthA] = a.month.split("-");
  const [yearB, monthB] = b.month.split("-");
  const weekA = a.week;
  const weekB = b.week;

  if (yearA !== yearB) return yearA - yearB;
  if (monthA !== monthB) return monthA - monthB;
  return weekA - weekB;
});
```

```
const drawChart = () => {
  const svg = d3.select(svgRef.current);
  svg.selectAll("*").remove(); // 이전 차트 지우기

  const width = 450; // 너비 450px
  const height = 350; // 높이를 350px로 설정
  const margin = { top: 40, right: 40, bottom: 40, left: 30 }; // 상단 여백을 늘림

  const labels = chartData.map(
    (item) => `${item.month.split("-")[1]}월 ${item.week}주차`
  );
  const applicationsData = chartData.map((item) => item.weeklyApplications);
  const cancellationsData = chartData.map((item) => item.weeklyCancellations);

  const x = d3
    .scaleBand()
    .domain(labels)
    .range([margin.left, width - margin.right])
    .padding(0.1);

  const y = d3
    .scaleLinear()
    .domain([0, 30]) // 최대 값을 데이터의 최대값보다 약간 높게 설정
    .range([height - margin.bottom, margin.top]);

  svg.attr("width", width).attr("height", height);
}

}

}
```

차량 관리 - 시각화 그래프



```
useEffect(() => {
  ...
  const data = response.data;
  if (!data || !Array.isArray(data)) {
    throw new Error("Invalid Data Structure");
  }

  const labels = data.map((item) => item.c_size || "Unknown Size");
  const usageData = data.map((item) => item.usageCount || 0);

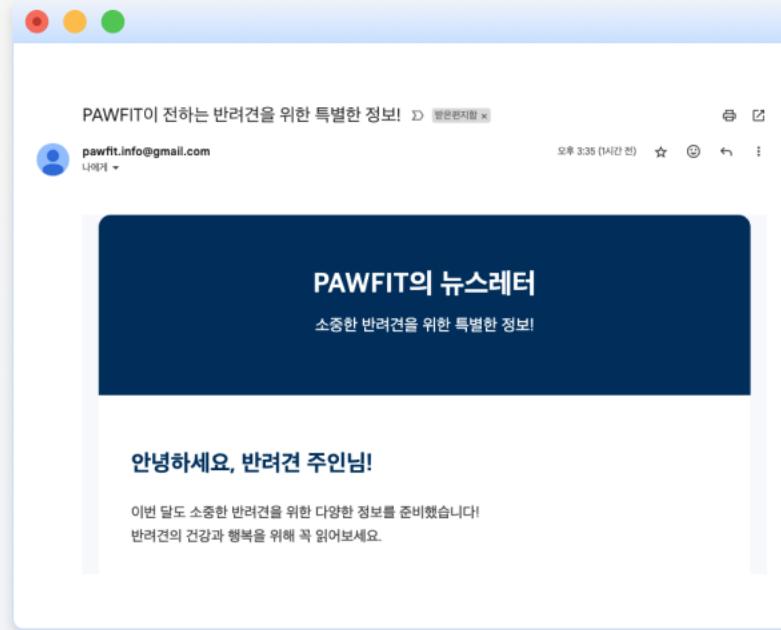
  setChartData({
    labels: labels,
    datasets: [
      {
        label: "Weekly Usage by Vehicle Size",
        data: usageData,
        backgroundColor: [
          ...
        }];
    return (
      <div>
        <Pie
          data={chartData}
          options={{
            plugins: {
              tooltip: {
                callbacks: {
                  label: (tooltipItem) => {
                    const label = tooltipItem.label || "";
                    const value = tooltipItem.raw || 0;
                    const total = tooltipItem.chart.data.datasets[0].data.reduce(
                      (a, b) => a + b,
                      0
                    );
                    const percentage = ((value / total) * 100).toFixed(2) + "%"; // 백분율 계산
                    return `${label}: ${percentage}`; // 라벨에 백분율 추가
                  },
                },
              },
            },
            dataLabels: {
              formatter: (value, context) => {
                const total = context.chart.data.datasets[0].data.reduce(
                  (a, b) => a + b,
                  0
                );
                const percentage = ((value / total) * 100).toFixed(2) + "%"; // 백분율 계산
                return percentage; // 데이터 레이블에 백분율 표시
              },
            }
          }>
    );
  }
});
```

CHART.JS



- CHART.JS 활용 차량의 지역별 크기별 이용 수를 나타내는 그래프 구현
- 해당 시각화를 통해 고객 선호도를 파악 및 효율적인 차량 운용 가능

회원 관리 - 뉴스레터



```
// 이메일 전송 메서드
public void sendEmail(String to, String subject, String htmlContent) {
    MimeMessage message = mailSender.createMimeMessage();
    MimeMessageHelper helper;

    try {
        helper = new MimeMessageHelper(message, true);
        helper.setTo(to);
        helper.setSubject(subject);
        helper.setText(htmlContent, true); // HTML 형식으로 설정

        mailSender.send(message);
        System.out.println("메일 전송 성공: " + subject); // 성공 메시지 출력
    } catch (MessagingException e) {
        System.err.println("메일 전송 실패: " + e.getMessage()); // 실패 메시지 출력
    }
}
```



JAVA MAIL

- 회원 목록에서 회원 정보 목록 열람 가능
- 회원가입 시 뉴스레터 수신을 동의한 회원에 한해 뉴스레터 보내기 활성화

01

OVERVIEW

02

ARCHITECTURE

03

DATABASE

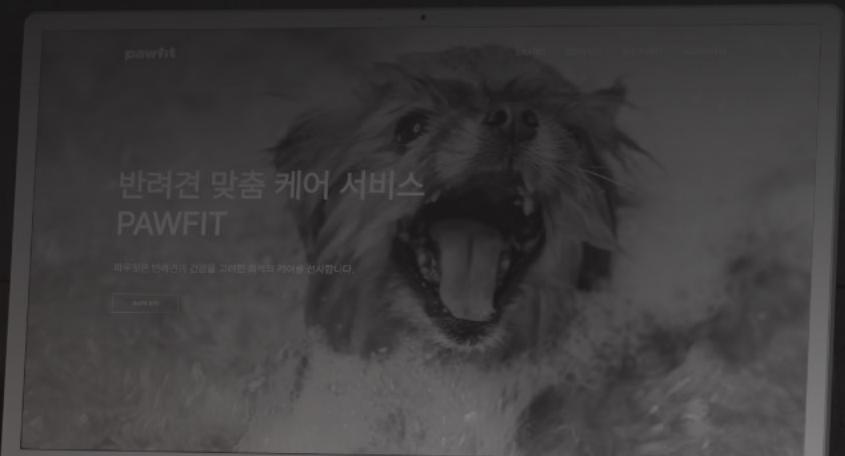
04

CODE

05

DEMONSTRATION

배포 및 시연



배포

The screenshot shows the Google Cloud VPC Firewall Rules interface. At the top, there's a message about port 25 being blocked. Below it, a note says to check the App Engine Firewall documentation. The main area is a table of rules:

이름	유형	대상	필터	프로토콜/포트	작업	우선순위	네트워크	로그
default-allow-http	인그레스	http-server	IP 범위: 0.0.0.0/0	tcp:80	허용	1000	default	사용 안함
default-allow-https	인그레스	https-server	IP 범위: 0.0.0.0/0	tcp:443	허용	1000	default	사용 안함
python1	인그레스	전체 적용	IP 범위: 0.0.0.0/0	tcp:8000	허용	1000	default	사용 안함
react	인그레스	전체 적용	IP 범위: 0.0.0.0/0	tcp:3000	허용	1000	default	사용 안함
spring	인그레스	전체 적용	IP 범위: 0.0.0.0/0	tcp:8080	허용	1000	default	사용 안함
default-allow-icmp	인그레스	전체 적용	IP 범위: 0.0.0.0/0	icmp	허용	65534	default	사용 안함
default-allow-internal	인그레스	전체 적용	IP 범위: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	허용	65534	default	사용 안함
default-allow-rdp	인그레스	전체 적용	IP 범위: 0.0.0.0/0	tcp:3389	허용	65534	default	사용 안함
default-allow-ssh	인그레스	전체 적용	IP 범위: 0.0.0.0/0	tcp:22	허용	65534	default	사용 안함



GOOGLE CLOUD

- 가상머신 및 방화벽 설정, 무료도메인 duckdns 사용
- 각 경로 요청 처리 / 클라이언트가 전송하는 파일 크기 등 설정 작성

```
server {
    listen 80;
    server_name pawfit.duckdns.org;
    # HTTP 요청을 HTTPS로 리디렉션
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name pawfit.duckdns.org;

    ssl_certificate /etc/letsencrypt/live/pawfit.duckdns.org/fullchain.pem;      #ssl 인증서 경로
    ssl_certificate_key /etc/letsencrypt/live/pawfit.duckdns.org/privkey.pem; #ssl 개인키 경로
    client_max_body_size 20M;

    location /api {
        proxy_pass http://localhost:8080; # Spring Boot 주소
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_cache_bypass $http_upgrade;
    }

    location /login {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /predict {
        proxy_pass http://localhost:8080/predict;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /kakao/callback {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location / {
        root /home/jihun@411/app/build; # React 빌드 경로
        try_files $uri $uri /index.html;
    }
}
```

PWA

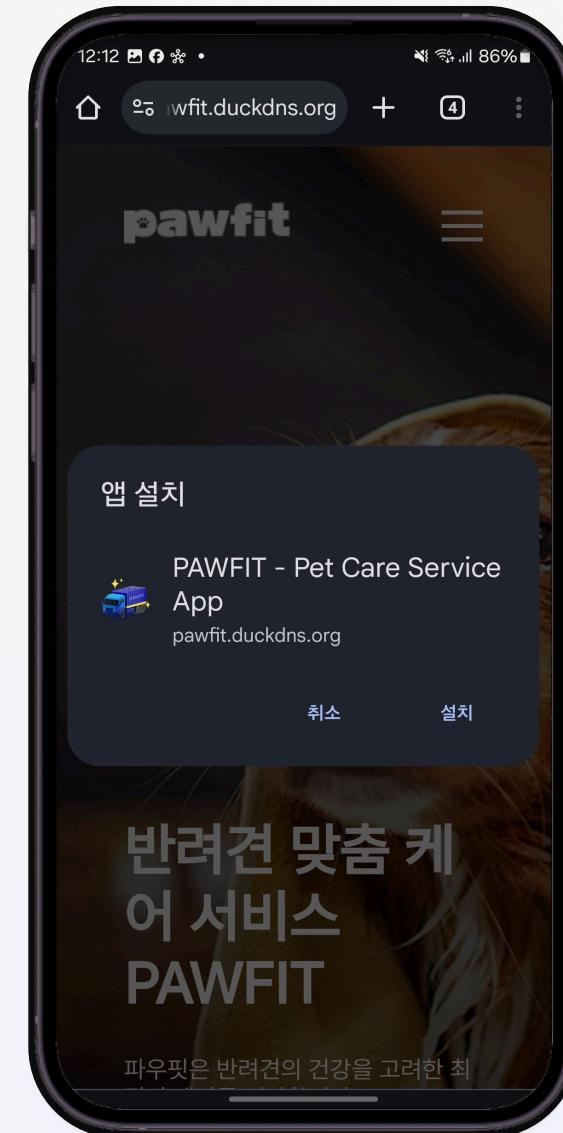
```
{  
  "short_name": "Pawfit",  
  "name": "PAWFIT - Pet Care Service App",  
  "icons": [  
    {  
      "src": "favicon.ico",  
      "sizes": "64x64 32x32 24x24 16x16",  
      "type": "image/x-icon"  
    },  
    {  
      "src": "logo192.png",  
      "type": "image/png",  
      "sizes": "192x192"  
    },  
    {  
      "src": "logo512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    }  
  "start_url": "/",  
  "display": "standalone",  
  "theme_color": "#000000",  
  "background_color": "#ffffff"  
}
```



PWA

- 리액트 프로젝트 생성 시 PWA 설정하여, 별도의 앱 개발 과정 없이 모바일 친화적 웹 개발이 가능

```
<!DOCTYPE html>  
<html lang="ko">  
  <head>  
    <meta charset="utf-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1" />  
    <meta name="theme-color" content="#000000" />  
    <meta name="description" content="PAWFIT - A pet care service app" />  
  
    <!-- Manifest 파일 링크 추가 -->  
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />  
  
    <!-- Favicon 설정 -->  
    <link rel="icon" href="/favicon.ico" />  
  
    <title>PAWFIT</title>  
  </head>  
  <body>  
    <div id="root"></div>  
  
    <!-- 서비스 워커 등록 -->  
    <script>  
      if ('serviceWorker' in navigator) {  
        window.addEventListener('load', () => {  
          navigator.serviceWorker  
            .register('/service-worker.js')  
            .then((registration) => {  
              console.log('ServiceWorker 등록 성공:', registration);  
            })  
            .catch((error) => {  
              console.error('ServiceWorker 등록 실패:', error);  
            });  
        });  
      }  
    </script>  
  </body>  
</html>
```



KOSMO 144기



pawfit

반려견을 위한 맞춤 케어를 원한다면, 파우핏

DEVELOPERS

김소미 하지현

김민진 김은하

박진수 임동규

정주호

