

인공지능을 통한 차량 사고 분석 및 견적 입찰 시스템

팀 명 : 차박GO
지도 교수 : 이병천 교수님
팀 장 : 여승철
팀 원 : 김찬일
김태현
정석현
조영무

2024. 10.

중부대학교 정보보호학과

목 차

1. 서론	
1.1 연구 배경	4
1.1 연구 목적 및 주제선정	4
2. 관련연구	
2.1 AI	
2.1.1 Python	5
2.1.2 Flask	5
2.1.3 U-net	5
2.1.4 VGG16	6
2.1.5 Cosine Similarity	6
2.1.6 Ngrok	6
2.2 BACKEND	
2.2.1 Java	7
2.2.2 Spring Boot	7
2.2.3 JPA	7
2.2.4 QueryDSL	8
2.2.5 MySQL	8
2.2.6 Redis	8
2.2.7 Spring Security	8
2.2.8 Oauth2	8
2.2.9 JWT	9
2.2.10 AWS EC2	9
2.2.11 AWS S3	9
2.2.12 AWS CloudFront	9
2.2.13 Swagger	9
2.3 FRONTEND	
2.3.1 Next.js	10
2.3.2 TypeScript	10
2.3.3 PNPM	10
2.3.4 ESLint, Prettier	10
2.3.5 Responsive Web	11
2.3.6 Zustand	11

2.3.7 Axios	11
2.3.8 React Query	11
2.3.9 CORS	12
2.3.10 CSP	12
2.3.11 Express.js	12
2.3.12 Docker	12
3. 본 론		
3.1 서비스 구상	13
3.2 개발 환경	13
3.3 아키텍처 구상도	14
4. 서비스 안내		
4.1 앱/웹 서비스 구성	15
4.2 유튜브 영상 QR	24
5. 서비스 안내		
5.1 결론 및 기대효과	25
5.2 추후 보완사항	25
6. 별첨		
6.1 참고 문헌	26
6.2 팀원 소개	26
6.3 소스 코드	26
6.4 시연영상 링크	27
6.5 발표 자료	27

1. 서론

1.1 연구 배경

현대사회에서 도시화와 교통 인프라의 발전으로 차량 운행량이 증가함에 따라 지난 몇 년간 차량 사고의 발생 건수가 증가하고 있지만, 사고 처리의 복잡성에 관한 개선이 이루어지지 않는다. 이러한 통계는 차량 사고가 더이상 개인적 문제를 넘어 사회적 문제로 자리잡고 있음을 시사한다. 사고 발생 후, 피해자는 신속하게 손상 정도를 평가하고 수리 비용을 산정해야 하며, 해당 과정은 전문 인력의 도움을 필요로 한다. 그러나 현재 보편적인 차량 손상 평가 방식은 주관성을 띠고 있으며, 처리 과정이 복잡하여 많은 시간과 비용을 소모하게 된다. 특히, 손상 평가 시 인력의 경험과 판단에 의존하는 경우가 많아 부정확하고 불공정한 평가가 이뤄질 수 있다. 이러한 문제를 해결하기 위해 인공지능(AI) 기술도입의 필요성이 확산되고 있다. 이를 활용하여 사고 차량의 파손 부위와 종류를 자동으로 분석하고 견적서를 신속하게 산출하여 차주 및 보험사의 부담을 경감 시킬 수 있다.

1.2 연구 목적 및 주제선정

차주는 차량 사고 발생 후, 사고 평가를 위해 전문가의 현장 방문, 견적 산출, 보험사와의 협의 등 다소 복잡한 단계를 거쳐야 하기 때문에 차주 및 보험사에 많은 부담을 요구한다. 이러한 과정에서 AI를 활용함으로써 대량의 데이터를 학습하여 미세한 손상까지도 정확히 판별할 수 있으며, 이는 기존 손상 평가 방식의 문제점 중 하나인 주관성을 최소화하는 데 기여한다. 또한 사용자가 단순히 사진을 찍어 업로드함으로써 손상 분석과 견적 산출이 자동으로 이루어지기 때문에 고객은 시간과 부담을 경감할 수 있다. 더불어, 보험 처리 절차의 효율성을 높이기 위해 기존의 복잡한 절차를 간소화하여 보험사와의 협의를 보다 원활하게 진행할 수 있게 한다. 또한 타 서비스와 다르게 커뮤니티 서비스를 통해 고객 간의 공업사에 관한 리뷰 형식이나 정보 제공을 통해 보다 원활한 소통을 할 수 있도록 지원한다. 이러한 커뮤니티 서비스는 고객들이 서로의 경험을 공유하게 하여, 공업사 선택 시 보다 객관적이고 신뢰할 수 있는 정보를 제공받을 수 있도록 한다. 궁극적으로, 인공지능 기술을 통해 차량 사고 처리의 비효율성을 해소하고, 고객의 경험을 개선하는 것을 목표로 하여 자동차 산업 전반에 긍정적인 영향을 주는 것을 목표로 하고 있다.

2. 관련 연구

2.1 AI

2.1.1 Python

Python은 고수준의 범용 프로그래밍 언어로, 가독성이 높고 문법이 간결하여 다양한 개발 환경에서 널리 사용된다. Python은 풍부한 표준 라이브러리와 서드파티 모듈을 제공하여 데이터 분석, 웹 개발, 인공지능, 자동화 등 여러 분야에서 활용 가능하다. 특히, 동적 타이핑과 자동 메모리 관리 기능을 갖추고 있어 개발 생산성을 높이는 장점이 있다. 해당 프로젝트에서는 Flask를 사용하여 웹 API 서버를 구축하였으며, 이미지 데이터를 전처리하고 예측 모델을 사용하는 과정에서 OpenCV(cv2)와 TensorFlow 등의 라이브러리를 활용하였다.

2.1.2 Flask

Flask는 Python 기반의 경량 웹 프레임워크로, 간결한 설계와 유연성을 제공하여 소규모 웹 애플리케이션이나 API 서버를 구축하는 데 적합하다. 최소한의 코드로 빠르게 웹 서비스를 개발할 수 있어 많은 개발자에게 선호되며, 필요에 따라 확장 가능한 구조를 가진다. 해당 프로젝트에서는 Flask를 사용하여 웹 API 서버를 구축하였으며, 클라이언트와 서버 간의 데이터 전송을 처리하였다. 또한, 프로젝트 중 발생한 CORS(Cross-Origin Resource Sharing) 문제를 해결하기 위해 flask-cors 라이브러리를 사용하여 외부 도메인으로부터의 요청을 안전하게 처리할 수 있도록 설정하였다. 이를 통해 이미지 데이터를 서버로 전송하고, 서버에서 예측 결과를 반환하는 통신이 원활하게 이루어졌다.

2.1.3 U-net

U-Net은 주로 이미지 분할 작업에 사용되는 딥러닝 모델로, 의료 영상 분석과 같은 세밀한 경계가 필요한 작업에서 뛰어난 성능을 발휘한다. U-Net은 대칭적인 U 형태의 구조를 가지고 있으며, 인코더(축소 경로)와 디코더(확장 경로)로 이루어져 있다. 인코더는 이미지의 특징을 추출하고, 디코더는 추출된 특징을 기반으로 원본 이미지의 해상도를 복원하여 각 픽셀에 대한 정확한 예측을 제공한다. 이 과정에서 다양한 해상도의 피쳐 맵을 결합하여 이미지 내의 세부적인 정보를 보존한다는 특징이 있다. 해당 프로젝트에서는 자동차 이미지의 파손 부위를 분석하기 위해 U-Net 알고리즘을 적용하였다. U-Net을 통해 이미지 내에서 파손된 부분을 분할하고, 이를 시각적으로 식별할 수 있도록 정확한 예측을 수행하였다. 이 방식은 자동차 손상 부위를 탐지하고 분석하는 데 효과적인 결과를 제공하였다.

2.1.4 VGG16

VGG16은 Oxford University's Visual Geometry Group에서 개발한 딥러닝 기반의 이미지 분류 모델로, ImageNet 데이터셋에서 매우 높은 성능을 보인 모델 중 하나이다. VGG16은 깊은 네트워크 구조(16개의 층)를 가지고 있으며, 비교적 단순한 3x3 필터를 사용하는 합성곱 층과 풀링 층을 반복적으로 쌓아 이미지의 시각적 패턴을 효과적으로 학습한다. 이 모델은 다양한 종류의 이미지를 인식할 수 있는 강력한 특징 추출 능력을 가지고 있어, 이미지 분류, 객체 탐지 등 여러 응용 분야에서 널리 사용된다. 해당 프로젝트에서는 VGG16 모델을 사용하여 이미지의 특징 값을 추출하였다. 사전 학습된 VGG16 모델을 통해 입력된 이미지의 시각적 패턴을 벡터 형태로 변환하고, 이를 기반으로 코사인 유사도 등의 방법으로 이미지 간 유사성을 판별하였다. 이를 통해 자동차 파손 부위 이미지에서 유사한 손상 패턴을 가진 이미지를 찾아내는 데 효과적으로 활용되었다.

2.1.5 Cosine Similarity

Cosine Similarity는 두 벡터 간의 유사도를 측정하는 방법으로, 벡터의 크기보다는 방향에 초점을 맞추어 유사성을 계산한다. 두 벡터가 이루는 각도의 코사인 값을 기반으로 하며, 두 벡터의 방향이 완전히 같으면 1, 서로 직교하면 0, 반대 방향이면 -1의 값을 갖는다. 벡터의 크기는 무시하고, 오직 벡터 간의 각도에만 의존하기 때문에 주로 텍스트 분석, 이미지 특징 비교와 같은 다양한 분야에서 사용된다. 해당 프로젝트에서는 자동차 파손 이미지 간 유사도를 평가하기 위해 Cosine Similarity를 사용하였다. VGG16 모델을 통해 추출된 이미지의 특징 벡터들 간의 유사도를 비교하여, 파손 패턴이 유사한 이미지를 찾아내는 데 활용되었다. 이를 통해 크기나 밝기 등의 차이에 영향을 받지 않고, 손상 형태의 유사성을 정확하게 판별할 수 있었다.

2.1.6 Ngrok

Ngrok은 로컬에서 실행되는 웹 서버나 애플리케이션을 인터넷에 노출할 수 있는 터널링 도구이다. 로컬 서버에 임시로 외부에서 접근 가능한 URL을 생성하여 개발자가 인터넷을 통해 자신의 로컬 환경을 테스트하거나 다른 사람과 공유할 수 있도록 한다. 이 도구는 특히 로컬에서 개발 중인 API, 웹 애플리케이션 등을 외부와 빠르게 연결할 때 유용하다. Ngrok은 HTTP 및 HTTPS 요청을 지원하며, 트래픽 로그와 실시간 요청 분석 등의 기능도 제공한다. 해당 프로젝트에서는 Flask 기반 웹 애플리케이션을 외부에서 테스트할 수 있도록 Ngrok을 사용하여 로컬 서버를 인터넷에 공개하였다. 이를 통해 외부 클라이언트가 로컬 환경에서 실행 중인 웹 API에 접근할 수 있게 하여, 개발 및 테스트 과정에서 편리성을 높였다.

2.2 BACKEND

2.2.1 Java

자바(Java)는 썬 마이크로시스템즈에서 1995년에 개발한 객체 지향 프로그래밍 언어이다. 플랫폼 독립성을 보장하는 특징을 갖고 있으며 "Write Once, Run Anywhere" 철학에 따라 한 번 작성된 코드는 JVM(Java Virtual Machine)만 설치되어 있으면 어떤 운영체제에서도 실행될 수 있다. 자바는 안정성과 보안성을 중시하여 대규모 애플리케이션부터 모바일, 웹 개발까지 다양한 분야에서 사용되며, 풍부한 라이브러리와 강력한 멀티스레딩 기능을 제공한다. 또한 정적 타입 언어로 컴파일 시 오류를 미리 발견할 수 있어 유지보수가 용이하고, 강력한 커뮤니티와 지원 덕분에 기업 환경에서도 널리 채택된다.

2.2.2 Spring Boot

스프링 부트(Spring Boot)는 스프링 프레임워크를 기반으로 한 경량화된 애플리케이션 개발 프레임워크로, 복잡한 설정을 최소화하고 빠르게 애플리케이션을 구축할 수 있도록 도와준다. 개발자는 필요한 의존성만 지정하면 스프링 부트가 자동 설정(Auto Configuration)을 통해 환경을 구성해 주며, 내장 웹 서버(예: Tomcat, Jetty)를 포함하고 있어 별도의 서버 설정 없이 독립 실행형 애플리케이션을 쉽게 만들 수 있다. 또한 마이크로서비스 아키텍처에 적합한 유연성과 다양한 스타터(Starter) 패키지를 제공해 REST API, 데이터베이스 연동, 보안 등 다양한 기능을 빠르게 구현할 수 있다. 이로 인해 스프링 부트는 생산성을 높이고 유지보수를 용이하게 하여, 웹 애플리케이션 및 마이크로서비스 개발에 널리 사용된다.

2.2.3 JPA

JPA(Java Persistence API)는 자바 애플리케이션에서 객체와 관계형 데이터베이스 간의 매핑을 지원하는 ORM(Object-Relational Mapping) 표준 인터페이스이다. 개발자는 데이터베이스 테이블을 객체(Entity)로 매핑하여, SQL을 직접 작성하지 않고도 데이터베이스 작업을 수행할 수 있으며, 이를 통해 객체와 데이터베이스 간의 패러다임 불일치를 해소한다. JPA는 데이터베이스 접근을 추상화하여 유지보수를 용이하게 하고, 코드 재사용성을 높여준다. JPA는 기본적인 CRUD 작업부터 복잡한 쿼리 생성까지 다양한 기능을 제공하여 대규모 애플리케이션에서도 효율적인 데이터 관리를 가능하게 한다.

2.2.4 QueryDSL

QueryDSL은 하이버네이트 쿼리 언어(HQL: Hibernate Query Language)의 쿼리를 타입에 안전하게 생성 및 관리해주는 프레임워크이다. QueryDSL은 정적 타입을 이용하여 SQL과 같은 쿼리를 생성할 수 있게 해 준다. 자바 백엔드 기술은 Spring Boot와 Spring Data JPA를 함께 사용한다. 하지만, 복잡한 쿼리, 동적 쿼리를 구현하는 데 있어 한계가 있다. 이러한 문제점을 해결할 수 있는 것이 QueryDSL이다. QueryDSL은 자바 코드로 SQL 문을 작성할 수 있어 컴파일 시에 오류를 발생하여 잘못된 쿼리가 실행되는 것을 방지할 수 있다.

2.2.5 MySQL

MySQL은 오픈소스 관계형 데이터베이스 관리 시스템이다. 관계형 데이터베이스 이기에 행과 열로 구성된 테이블에 데이터를 저장하며 구조화된 쿼리 언어인 SQL을 이용하여 데이터를 정의, 조작, 제어, 쿼리 할 수 있다.

2.2.6 Redis

Redis는 Remote Dictionary Server의 약자로서, "키-값" 구조의 비정형 데이터를 저장하고 관리하기 위한 오픈 소스 기반의 비관계형 데이터베이스 관리 시스템이다. 모든 데이터를 메모리에서 처리하는 인메모리 데이터베이스로, 지연 시간이 매우 낮아 고속 처리가 요구되는 캐싱, 세션 관리, 실시간 분석 등에 적합하다. 메모리 기반의 특성상 휘발성이 있으므로, 메모리에 있는 데이터 전체에서 스냅샷을 작성하고, 이를 디스크에 저장하는 RDB(Redis Database) 방식 백업과 데이터가 변경되는 이벤트가 발행하면 이를 모두 로그에 저장하는 AOF 방식 백업을 제공한다.

2.2.7 Spring Security

Spring Security는 Spring 기반 애플리케이션의 인증, 인가 및 보안을 담당하는 스프링 하위 프레임워크이다. Spring Security는 인증(Authentication)과 권한(Authorization)에 대한 부분을 Filter 기반으로 동작하고 Spring MVC와 분리되어 관리 및 동작하며, 데이터 보호 기능을 포함하여 웹 개발 과정에서 필수적인 사용자 관리 기능을 구현하는데 도움을 준다.

2.2.8 OAuth2

OAuth 2.0은 인증에 대한 산업 표준 프로토콜로, 제 3자 애플리케이션이 사용자를 대신해 HTTP 서비스를 이용할 수 있는 권한을 부여하는 메커니즘을 제공한다. 이 프로토콜은 사용자의 로그인 정보를 제 3자 앱이나 웹사이트와 공유하지 않고도, 해당 앱이나 웹사이트가 사용자의 대신에 특정 작업을 수행하도록 허용한다.

2.2.9 JWT

JWT(JSON Web Token)란 인증에 필요한 정보들을 암호화시킨 JSON 토큰을 의미한다. 그리고 JWT 기반 인증은 JWT 토큰(Access Token)을 HTTP 헤더에 실어 서버가 클라이언트를 식별하는 방식이다.

2.2.10 AWS EC2

AWS EC2(Amazon Elastic Compute Cloud)는 AWS에서 제공하는 확장 가능하고 유연한 가상 서버 호스팅 서비스이다. 사용자는 EC2를 통해 필요에 따라 다양한 유형의 인스턴스를 생성하고 관리할 수 있으며, 이러한 인스턴스는 다양한 운영 체제와 애플리케이션 환경을 지원한다. EC2는 수요에 따라 자동으로 인스턴스를 확장하거나 축소할 수 있는 기능을 제공하여, 트래픽 변화에 맞춰 비용 효율적으로 자원을 관리할 수 있다.

2.2.11 AWS S3

S3는 AWS에서 제공하는 클라우드 객체 스토리지 서비스이다. 우수한 확장성, 데이터 가용성, 보안 및 성능을 제공하며, 대규모 데이터를 안전하고 확장 가능하며 비용 효율적으로 저장할 수 있다. 사용자는 IAM(Identity and Access Management)을 통해 특정 사용자나 그룹에 대한 권한을 세밀하게 조정하고, 리소스 기반 정책을 통해 S3 버킷 및 객체에 대한 액세스를 제어할 수 있다. IAM을 사용하면 사용자 인증과 권한 부여를 통해 안전한 데이터 관리가 가능하다.본 프로젝트에서는 S3에 이미지를 업로드하여 이미지 파일 저장 스토리지로써 활용하였다.

2.2.12 AWS CloudFront

CloudFront는 AWS의 CDN(Content Delivery Network) 서비스이다. CDN 서비스란 지리적 제약 없이 전 세계 사용자에게 빠르게 콘텐츠를 전송하는 서비스로 서버와 사용자 사이의 물리적인 거리를 줄여 콘텐츠 로딩에 소요되는 시간을 최소화한다. 본 프로젝트에서는 원활한 이미지 요청 처리와 S3 직접 접근 차단 및 이미지 url에 S3 버킷 이름과 지역이 노출되어 이를 보완하기 위해 사용하였다.

2.2.13 Swagger

Swagger는 RESTful API를 설계, 구축, 문서화, 테스트할 수 있도록 도와주는 오픈 소스 프레임워크이다. Swagger는 API의 구조를 정의하는 표준 포맷인 OpenAPI Specification(OAS)을 기반으로 하며, Swagger UI를 통해 API 문서를 시각적으로 제공하며, 사용자는 이 UI에서 API의 엔드포인트를 쉽게 탐색하고 테스트할 수 있다. 명확한 문서화와 테스트 기능을 통해 API 개발 주기를 단축하고, 협업을 개선하여 전체적인 개발 품질을 향상시키는 데 기여한다.

2.3 FRONTEND

2.3.1 Next.js

Next.js는 리액트를 기반으로 하는 프레임워크로, 서버사이드 렌더링(SSR)과 정적 사이트 생성(SSG)을 지원하여 웹 애플리케이션의 성능과 SEO(검색 엔진 최적화)를 크게 향상시킨다. 또한, 자동 코드 분할을 통해 페이지 로딩 속도를 최적화하며, 라우팅 및 API 엔드포인트 기능을 내장하여 개발 생산성을 높인다. 기본적으로 파일 시스템을 기반으로 하는 라우팅을 제공하여 설정이 간단하고, 복잡한 설정 없이도 빠르게 프로덕션 수준의 웹 애플리케이션을 구축할 수 있다. 따라서 SEO와 사용자 경험을 중시하는 웹 서비스에서 주로 사용된다.

2.3.2 TypeScript

TypeScript는 자바스크립트에 정적 타입 시스템을 도입한 언어로, 코드의 안정성과 유지보수성을 높이는 역할을 한다. 타입을 명시적으로 정의할 수 있어 컴파일 단계에서 오류를 사전에 발견할 수 있으며, 이는 런타임 오류를 줄여준다. 또한, 코드 자동 완성 기능과 타입 체크를 지원함으로써 개발자 경험을 향상시키고, 대규모 프로젝트에서 명확한 인터페이스와 구조를 정의할 수 있어 협업에도 유리하다. 자바스크립트와 완벽하게 호환되며, 기존 코드에 점진적으로 타입을 적용할 수 있는 유연성을 제공한다.

2.3.3 PNPM

PNPM은 Node.js 환경에서 사용되는 패키지 매니저로, 의존성 설치 속도가 빠르고 디스크 공간을 절약하는 특징을 갖고 있다. 기존 패키지 매니저인 NPM이나 Yarn과 달리, 중복된 패키지를 하나의 저장소에서 공유하도록 하여 디스크 사용량을 줄이며, 패키지를 설치할 때 하드 링크를 사용해 의존성 관리를 효율적으로 수행한다. PNPM은 모노레포 구조에서 특히 유용하며, 프로젝트 내 여러 패키지를 간편하게 관리할 수 있는 강력한 도구로 자리잡고 있다.

2.3.4 ESLint, Prettier

ESLint는 자바스크립트 코드의 문법을 검사하고 일관된 스타일을 유지하도록 돕는 린터 도구이다. 코드 품질을 유지하기 위해 사용되는 규칙들을 정의하고, 코드 작성 시 발생할 수 있는 잠재적인 오류를 사전에 발견할 수 있도록 한다. Prettier는 코드 포매팅 도구로, 자동으로 일관된 코드 스타일을 적용하여 가독성을 높인다. ESLint와 Prettier를 함께 사용함으로써 코드 품질과 스타일을 모두 관리할 수 있으며, 협업 시 코드 스타일에 대한 불필요한 논쟁을 줄일 수 있다.

2.3.5 Responsive Web

반응형 웹은 사용자가 어떤 디바이스에서 접속하더라도 최적의 사용자 경험을 제공할 수 있도록 웹 페이지 레이아웃과 콘텐츠가 자동으로 조정되는 웹 디자인 방식이다. 화면의 크기, 해상도, 방향에 따라 유동적으로 레이아웃이 변하며, 모바일, 태블릿, 데스크탑 등 다양한 환경에서 일관되고 편리한 사용성을 제공한다. 주로 CSS의 미디어 쿼리(Media Query)를 사용하여 화면 크기에 따라 다른 스타일을 적용하며, 이미지나 그리드와 같은 요소들도 크기에 맞춰 유연하게 변경된다. 반응형 웹은 유지보수 측면에서 효율적이고, 사용자 이탈을 줄이며 SEO에도 긍정적인 영향을 미치기 때문에 현대 웹 개발에서 필수적인 요소로 자리잡고 있다.

2.3.6 Zustand

Zustand는 리액트 애플리케이션에서 전역 상태를 관리하기 위한 경량 상태 관리 라이브러리이다. 간결한 API를 제공하며, 사용하기 쉬운 상태 관리 방식으로 전역 상태를 다루는 것이 가능하다. 기존의 Redux에 비해 설정이 간단하고 보일러플레이트 코드가 적어 빠르게 상태 관리를 시작할 수 있다는 장점이 있다. 특히 중소 규모의 프로젝트나 비교적 단순한 상태 관리가 필요한 경우에 적합하며, 성능도 매우 우수하다. 상태를 효율적으로 관리하면서도 코드의 복잡성을 줄이는 데 도움을 준다.

2.3.7 Axios

Axios는 자바스크립트 기반의 HTTP 클라이언트로, 브라우저와 Node.js 환경에서 모두 사용 가능하다. REST API와의 상호작용을 쉽게 할 수 있도록 도와주며, 프로미스를 사용하여 비동기 요청을 처리할 수 있다. JSON 데이터를 자동으로 변환하거나, 요청 및 응답을 인터셉트하여 공통 작업을 처리할 수 있는 기능을 제공한다. 또한, 요청 취소와 같은 다양한 편의 기능을 제공하여 네트워크 통신 시의 복잡한 시나리오를 간단하게 구현할 수 있다.

2.3.8 React Query

React Query는 서버 상태 관리 라이브러리로, 데이터를 서버에서 효율적으로 가져오고 캐싱하며, 클라이언트 측에서 데이터를 사용할 수 있도록 돕는다. 데이터 페칭, 캐싱, 동기화, 업데이트 등 서버와 관련된 복잡한 로직을 간단하게 구현할 수 있게 해주며, UI가 데이터를 중심으로 자동으로 업데이트되도록 처리해 준다. 특히, 실시간으로 데이터를 갱신하거나 서버와 클라이언트 간의 상태를 동기화하는 작업에서 강력한 성능을 발휘하며, 사용자 경험을 개선하는 데 중요한 역할을 한다.

2.3.9 CORS

CORS는 웹 애플리케이션이 다른 도메인에서 리소스를 요청할 수 있도록 허용하는 보안 메커니즘이다. 기본적으로 웹 브라우저는 보안상의 이유로 동일 출처 정책(Same-Origin Policy)을 적용하여, 다른 출처의 리소스를 가져오는 것을 제한한다. 하지만, CORS를 통해 특정 도메인이나 리소스를 허용하도록 설정할 수 있다. 서버측에서 Access-Control-Allow-Origin 헤더를 사용하여 어떤 도메인에서 요청을 허용할지 명시할 수 있으며, 이를 통해 클라이언트는 해당 리소스를 정상적으로 가져올 수 있다. 이 메커니즘은 API와 클라이언트 간의 상호작용에서 필수적이며, 잘못된 설정 시 보안 취약점이 발생할 수 있어 주의가 필요하다.

2.3.10 CSP

CSP는 웹 사이트에서 악성 콘텐츠가 실행되는 것을 방지하기 위한 보안 정책으로, 스크립트 인젝션과 같은 공격으로부터 보호하기 위해 사용된다. 개발자는 CSP 헤더를 통해 웹 페이지에 허용되는 콘텐츠의 출처를 명시할 수 있으며, 이를 통해 신뢰할 수 없는 외부 소스의 코드 실행을 막을 수 있다. 예를 들어, 스크립트, 이미지, 스타일 시트 등의 출처를 명확히 정의함으로써 크로스사이트 스크립팅(XSS)과 같은 공격을 예방할 수 있다. CSP는 웹 애플리케이션의 보안을 강화하기 위해 매우 중요한 역할을 하며, 브라우저가 페이지를 로드하는 동안 허용되지 않은 리소스가 있을 경우 이를 차단하고 경고를 발생시킨다.

2.3.11 Express.js

Express.js는 Node.js 환경에서 웹 애플리케이션을 구축하기 위한 경량 웹 프레임워크로, 빠르고 간결한 서버 개발을 가능하게 한다. 다양한 HTTP 메서드와 미들웨어를 지원하여, 요청과 응답을 효율적으로 처리할 수 있는 유연성을 제공한다. Express는 REST API 구축에 적합하며, 라우팅을 간편하게 관리할 수 있어 웹 서비스나 백엔드 서버 개발에서 많이 사용된다. 또한, 수많은 서드파티 미들웨어를 제공하여 세션 관리, 데이터 파싱, 오류 처리 등 다양한 기능을 쉽게 확장할 수 있다. Express는 그 간결함과 확장성 덕분에 초보 개발자부터 전문가까지 폭넓게 사용되며, 다양한 규모의 프로젝트에서 널리 채택되고 있다.

2.3.12 Docker

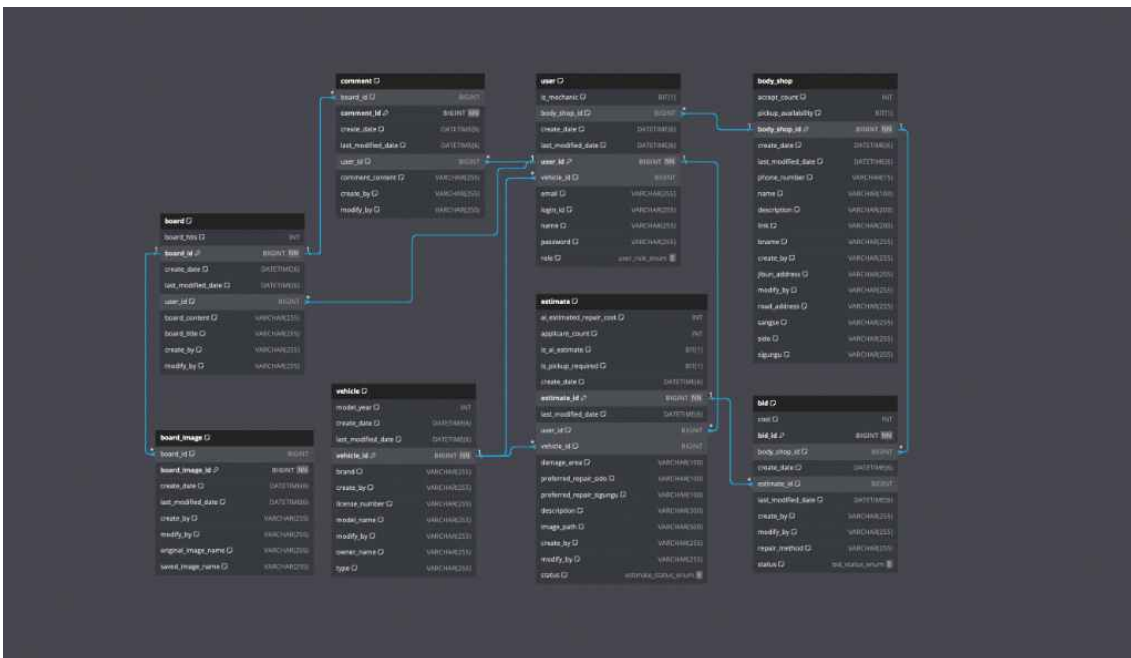
Docker는 애플리케이션을 가상 컨테이너 안에서 실행할 수 있도록 해주는 컨테이너 기반의 오픈 소스 플랫폼이다. 이 컨테이너는 운영체제(OS)와 독립적인 환경을 제공하여 애플리케이션과 그 종속성을 패키징함으로써, 어떤 환경에서도 일관되게 실행될 수 있도록 보장한다. Docker를 사용하면 개발, 테스트, 프로덕션 환경 간의 차이로 인한 문제를 최소화할 수 있으며, 환경 설정을 코드로 관리하여 DevOps 프로세스를 간소화하고 자동화를 촉진한다. 컨테이너는 가볍고 시스템

리소스를 효율적으로 사용하며, 독립적인 격리 환경을 제공하여 보안에도 장점이 있다. Docker는 마이크로서비스 아키텍처와도 잘 어울리며, 대규모 애플리케이션을 모듈화하고 배포하는 데 유리하다.

3. 본론

3.1 서비스 구상

그림 1은 본 서비스 DB의 ERD(Entity Relationship Diagram)이다.



[그림 1. 데이터베이스 ERD]

3.2 개발 환경

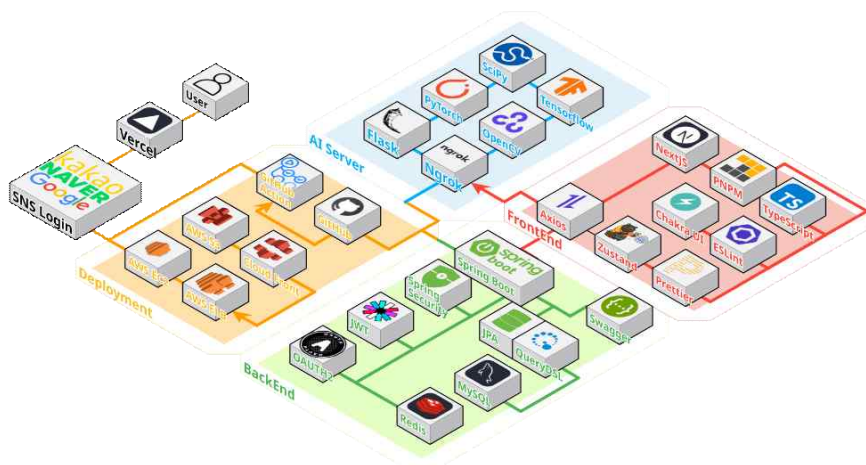
분류	내용
프로그래밍 언어	Java17, Python3.7(Conda), TypeScript(Node v20.14)
프레임워크/런타임	Spring Boot, Spring Security, JPA(QueryDSL), Flask, Next.js
데이터베이스	MySQL, Redis
인프라	AWS EC2, AWS S3, AWS CloudFront, AWS ELB, Ngrok
개발 도구	Intellij, Jupyter Notebook, Visual Studio Code, Github, Postman, Docker
협업 도구	Swagger, Github Issues, Notion, Discord, Draw.io

[표 1. 개발 환경]

3.3 아키텍처 구상도

그림 2는 전체적인 서비스 구상도에서 차량 손상 판정 및 견적 시스템의 아키텍처 구성을 나타낸다. 사용자는 카카오, 네이버, 구글과 같은 SNS 로그인을 사용하거나 직접 회원가입 및 로그인을 통해 서비스에 접근할 수 있다. 인증 과정에서 사용자의 로그인 요청은 Vercel에 배포된 프론트엔드에서 시작되며, 인증 정보는 JWT를 통해 Spring Security에서 관리된다. Access Token 만료 시에는 Redis에 저장된 Refresh Token을 사용해 Access Token을 재발급한다.

차량 손상 이미지는 사용자가 프론트엔드를 통해 업로드하며, 해당 이미지는 AWS S3 버킷에 저장된다. 이미지 저장 과정에서는 중복 방지를 위해 원본 파일명에 UUID가 결합된 파일명을 사용하며, 업로드된 이미지는 CloudFront를 통해 제공된다. CloudFront를 통해 이미지에 대한 접근 속도를 높이고, 직접 접근을 제한해 보안성을 강화하고 있다. 차량 손상 이미지 분석 요청은 Flask 서버를 통해 AI 서버로 전달된다. AI 서버는 PyTorch, TensorFlow, OpenCV 등의 라이브러리를 활용해 차량 파손 이미지를 분석하고, 예상 수리 비용 및 필요한 부품을 도출한다. 분석 결과는 REST API 형식으로 프론트엔드에 전달되어 사용자가 견적 정보를 확인할 수 있다. AI 서버는 이미지 분석 시 메모리 사용량이 급격히 증가하는 피크 현상이 발생하며, 이때 약 3GB~4GB의 메모리를 사용하게 된다. 이러한 메모리 부담을 줄이기 위해 로컬 환경에서 Ngrok을 사용해 외부 네트워크와 통신하도록 구성하였다. 사용자가 생성한 견적은 공개 설정이 가능하며, 공개된 견적은 여러 공업사로부터 입찰을 받을 수 있다. 공업사 사용자는 QueryDSL을 통한 동적 쿼리를 생성하여 다양한 조건으로 견적을 검색할 수 있으며, 반환된 검색 결과로 입찰을 신청할 수 있다. 입찰 요청은 "WAITING" 상태로 등록되며, 사용자가 원하는 공업사를 선택해 수락할 경우 입찰 상태는 "ACCEPTED"로 변경되고 최종적으로 수리 계약이 성사된다.



[그림 2. 아키텍처 구상도]

4. 서비스 안내

4.1 앱/웹 서비스 구성

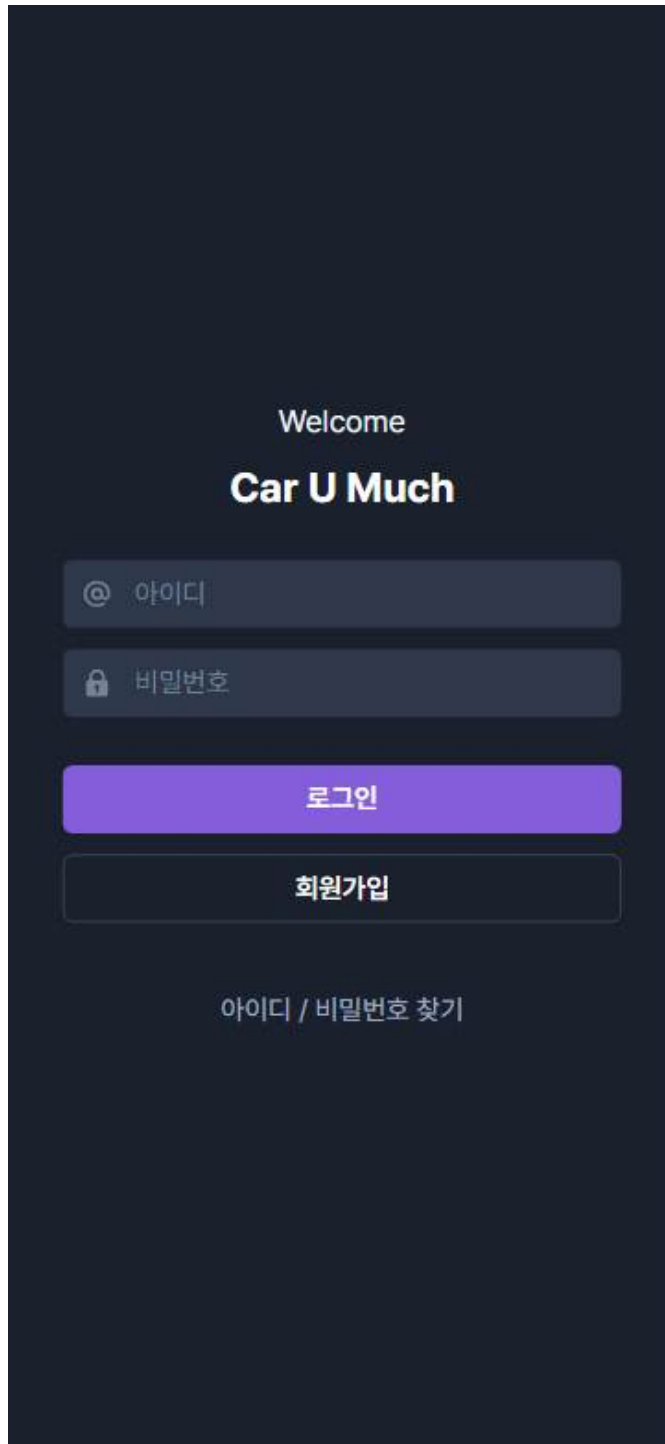
그림 3은 앱/웹 서비스의 초기 화면이다.



[그림 3. 앱/웹 서비스 초기 화면]

위 그림 3은 일반 로그인 외에도 SNS 로그인을 지원한다. 지원 방식은 네이버 로그인, 카카오 로그인, 구글 로그인으로 구성되어 있다.

그림 4은 일반 로그인 화면이다.



[그림 4. 일반 로그인 화면]

일반 로그인 화면에서 비밀번호 찾기 과정으로 서버가 인증번호를 생성하고 Redis의 TTL 기능을 통해 120초 동안 저장하여 사용자 이메일로 발송한다. 사용자가 인증번호를 입력하면 임시 토큰이 발급되어 새로운 비밀번호로 변경할 수 있는 절차가 진행된다.

그림 5은 회원가입 화면이다.

[그림 5. 회원가입 화면]

프론트엔드에서 정규표현식을 이용해서 1차로 아이디, 비밀번호 규칙을 검증하고, 2차로 백엔드에서 값을 검증한다. 프로젝트 내에서 발생하는 이벤트들은 Chakra UI 로 제작된 공용 모달 hook을 이용하여 표시된다.

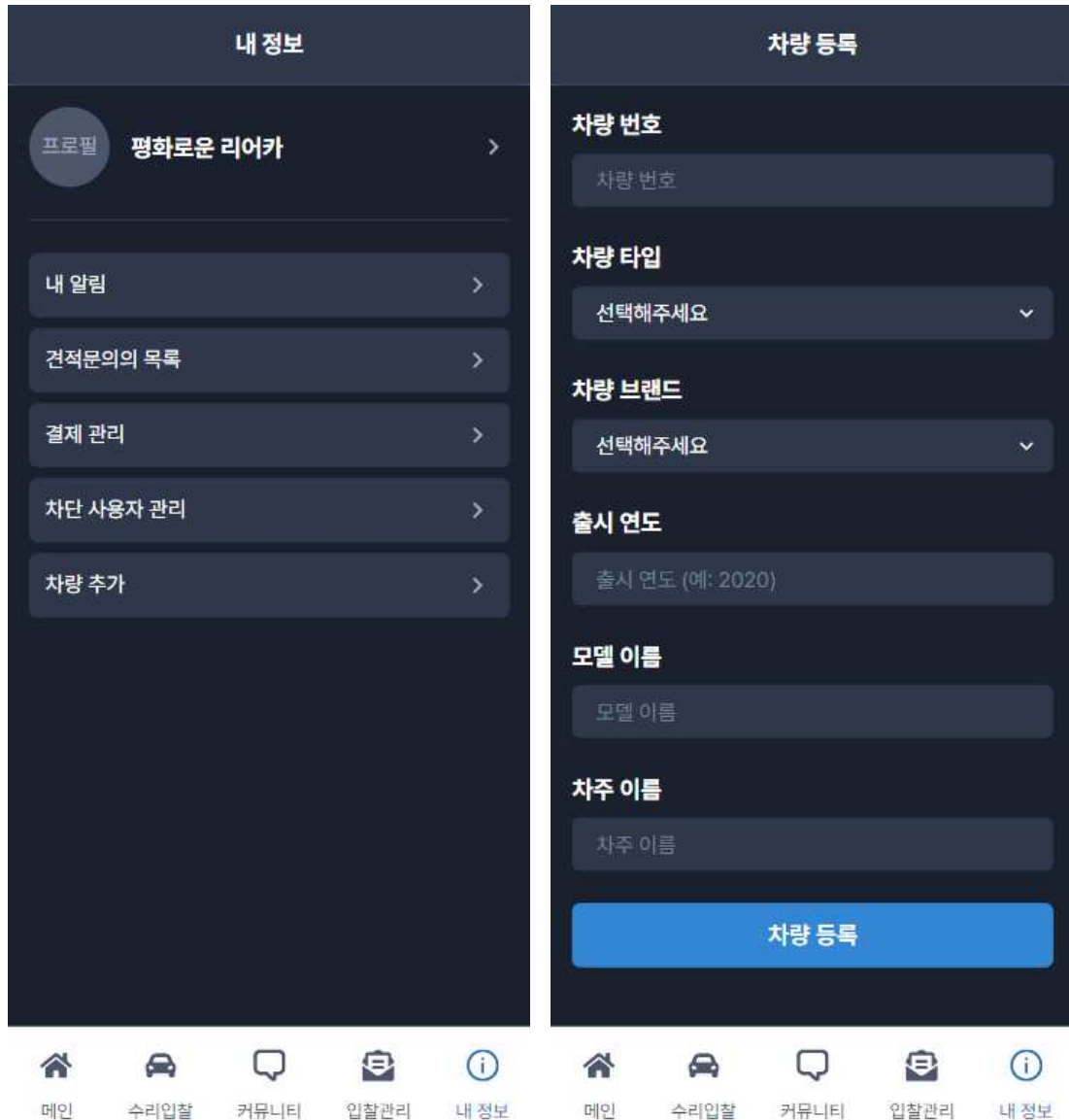
그림 6은 메인 화면이다.



[그림 6. 메인 화면]

메인 화면에는 내 차 등록, 견적받기, 커뮤니티, 공업사 등록, 최근 글 등의 기능이 있다.

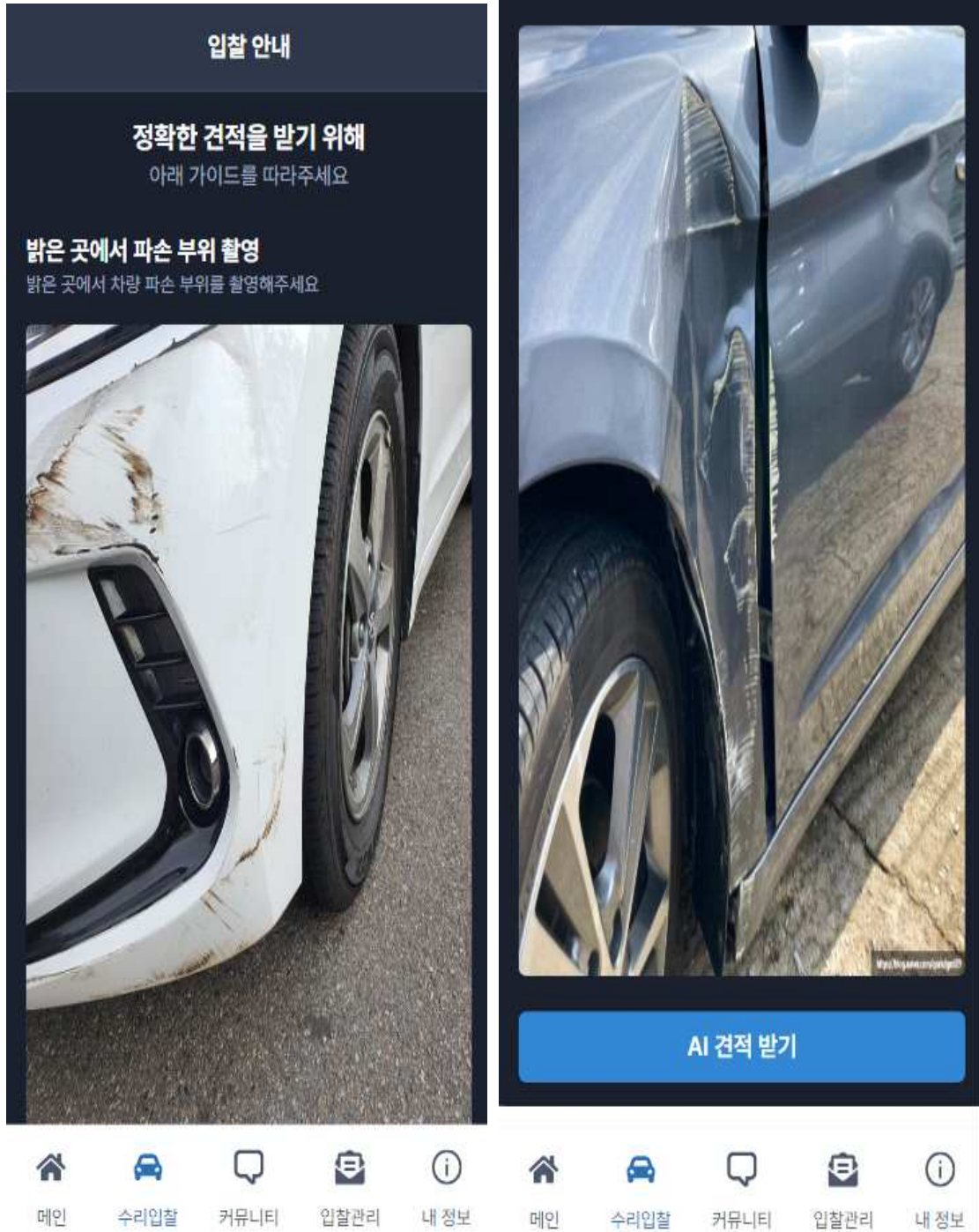
그림 7은 차량 등록 화면이다.



[그림 7. 차량 등록 화면]

공업사 회원이 아닌 사용자는 시건적을 받기 위해 내 정보에서 차량 추가를 해야 한다. 차량번호, 차량 타입, 차량 브랜드, 출시 연도, 모델 이름, 차주 이름 등을 작성하여 등록할 수 있다.

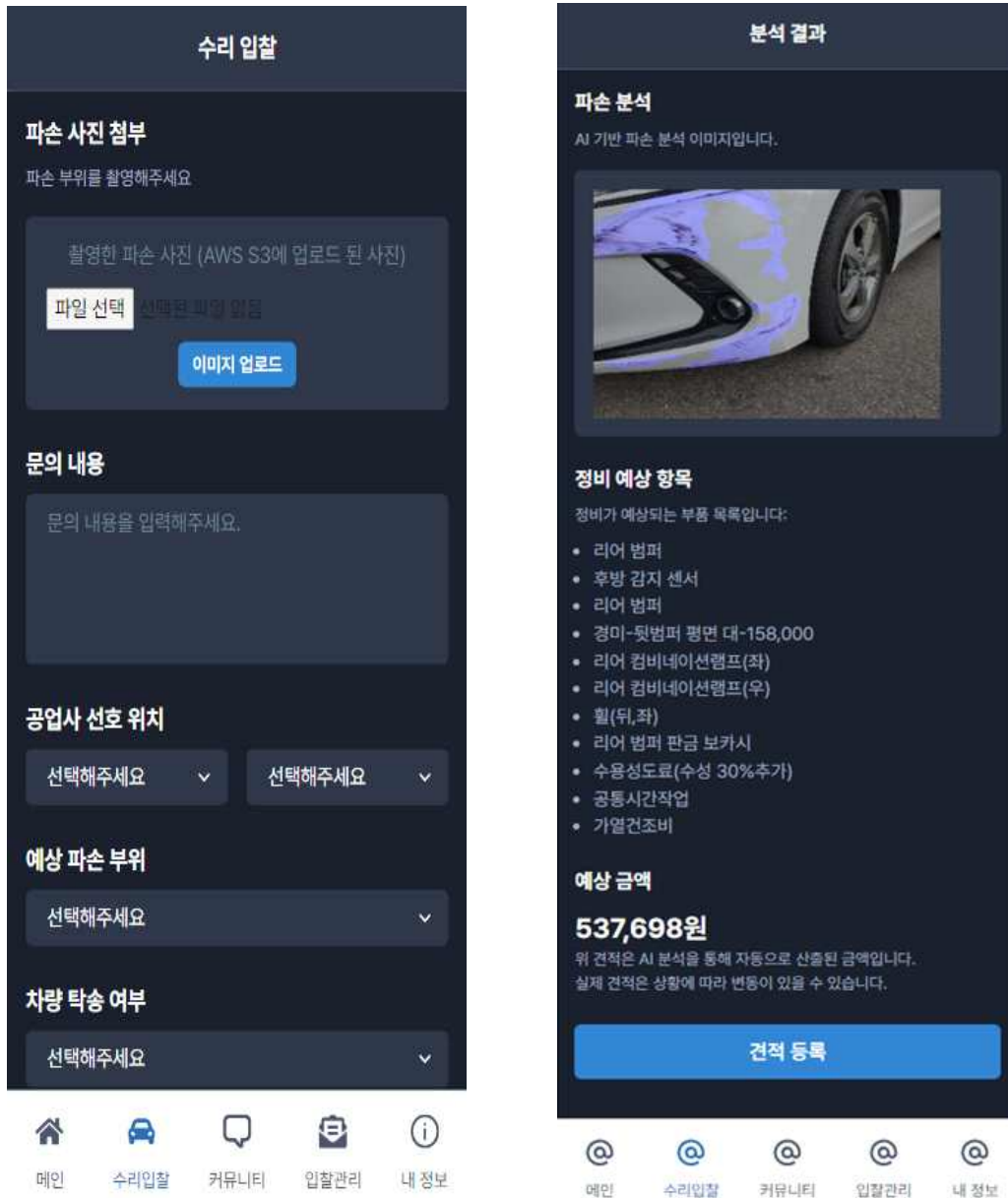
그림 8은 입찰 안내 화면이다.



[그림 8. 입찰 안내 화면]

수리입찰 가이드를 따라 공업사 회원이 아닌 사용자는 본인의 차량을 등록한 후 AI 견적 받기를 터치하면 수리 입찰 화면으로 이동하게 된다.

그림 9는 수리 입찰 화면이다.



[그림 9. 수리 입찰 화면]

사용자는 차량 사고 사진 및 문의내용, 공업사 선호 위치, 예상 파손 부위, 탁송 여부, 차량 브랜드 등을 체크 후 견적을 접수한다. 해당 과정에서 클라이언트가 이미지를 첨부하면 S3 내 이미지 파일 이름 중복 방지를 위해 이미지 파일 이름을 오리지널 이미지 파일 이름 + uuid 로 변경되고, Spring의 MultipartFile 인터페이스를 사용하여 CloudFront를 거쳐 S3에 이미지를 업로드하게 되며, CloudFront를 통해 접근하는 이미지 url을 반환한다. 견적 데이터를 저장한 뒤 최종적으로 입력한 정보 및 파손 사진을 기반으로 AI 서버가 분석하여 수리가 필요할 것으로 예상되는 부품 목록과 예상 수리비를 반환한다.

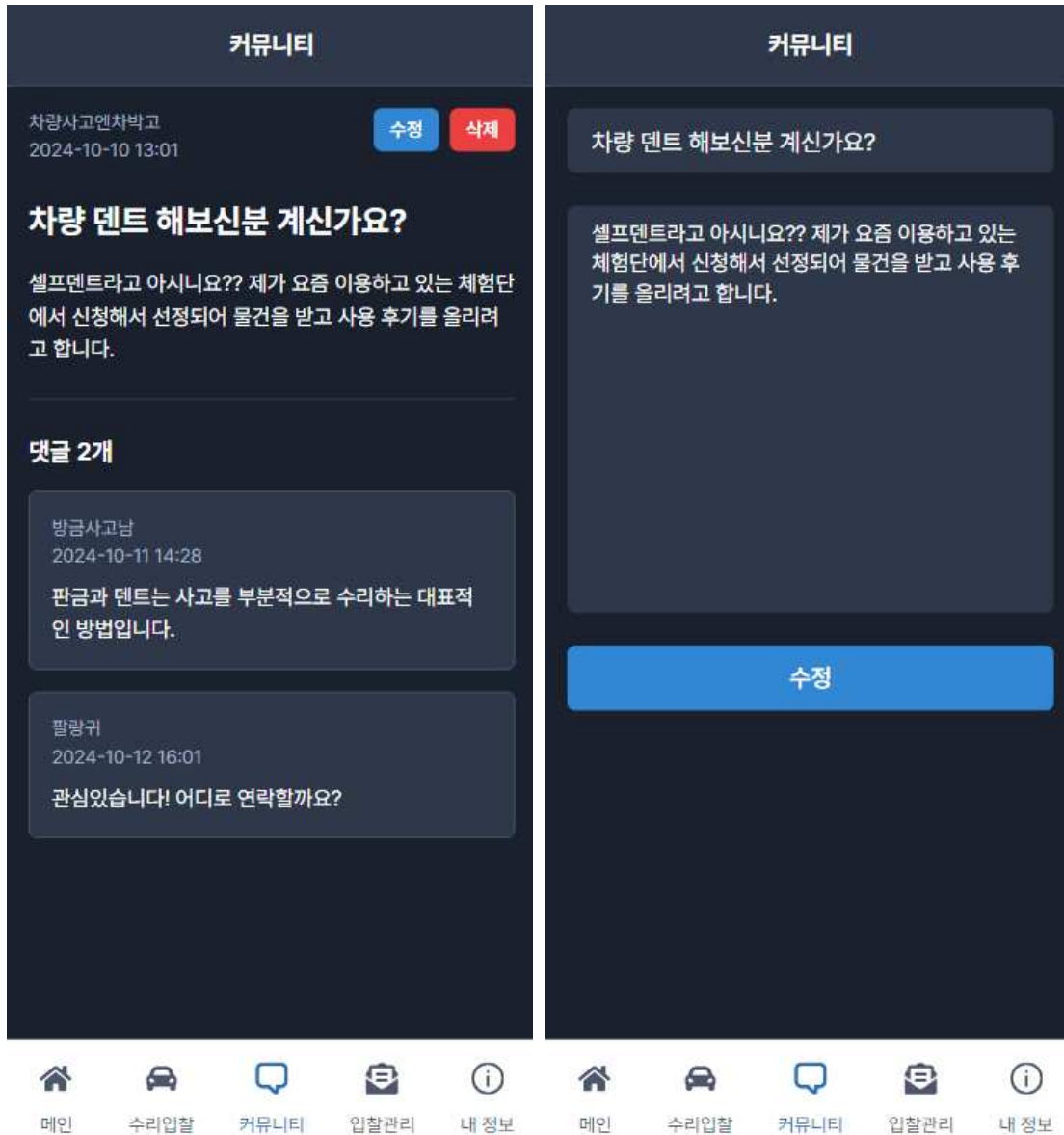
그림 10은 커뮤니티 화면이다.



[그림 10. 커뮤니티 화면]

커뮤니티는 일반적인 게시판 기능과 같이 사용자들이 의견을 나눌 수 있는 기능이다. 해당 기능을 통해 공업사에 대한 리뷰 및 정보 공유가 가능해짐으로써, 사용자들은 보다 객관적이고 합리적인 결정을 도출 할 수 있다.

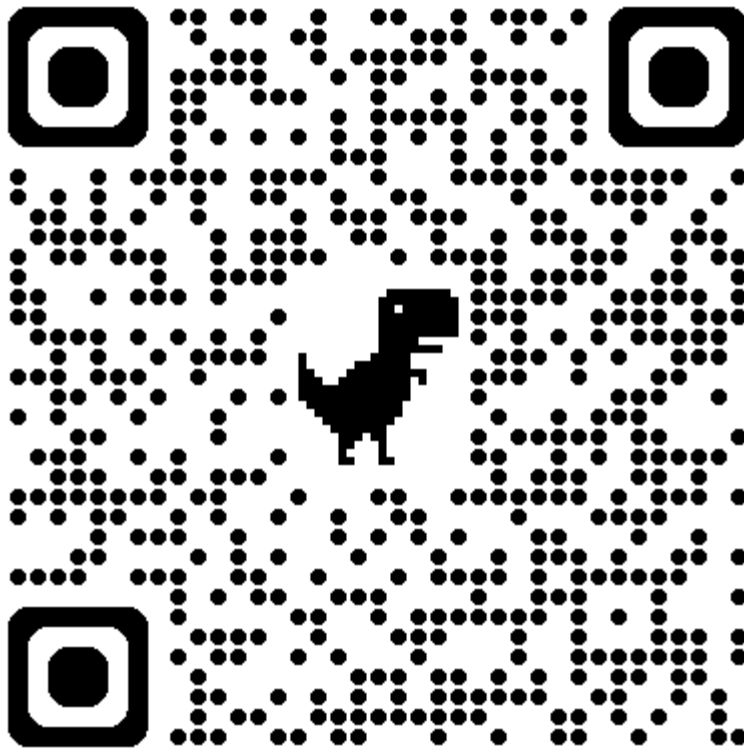
그림 11은 커뮤니티 글 작성 화면이다.



[그림 11. 커뮤니티 글 작성 화면]

커뮤니티 글에는 사용자들이 댓글을 작성할 수 있다. 수정을 터치시 본인이 작성한 글을 자유롭게 수정 삭제 할 수 있어 사용자들이 보다 쉽게 소통할 수 있도록 편의성과 접근성을 높이고자 하였다.

4.2 유튜브 영상 QR



[그림 12. 졸업작품 시연영상 유튜브 링크 QR코드]

주소 : <https://www.youtube.com/watch?v=xeeVluCMCA>

5. 결론

5.1 결론 및 기대효과

본 시스템은 차주가 사고 차량의 사진을 간편하게 업로드하면, 인공지능이 자동으로 손상을 분석하고 예상 수리 비용을 산출하여 신속하게 정보를 제공한다. 이를 통해 차주는 복잡한 절차를 간소화하고, 보다 빠르고 효율적으로 사고 처리를 진행할 수 있다. 또한, 고객 간의 소통을 원활하게 하는 커뮤니티 기능을 통해 공업사에 대한 리뷰 및 정보 공유가 가능해짐으로써, 고객들은 보다 객관적이고 합리적인 결정을 도출할 수 있다. 끝내 해당 시스템을 통해 보험사와 공업사 간의 협업을 강화하고 고객의 만족도를 높이며, 사고 처리 과정에서의 투명성 또한 확보할 수 있다.

5.2 추후 보완사항

본 연구에서 개발한 차량 사고 손상 분석 AI 모델은 VGG16 아키텍처를 활용하여 높은 정확성을 보여주었지만, 데이터의 한계와 모델 크기의 비효율성이 존재하였다. 사용하고 있는 데이터의 양이 한정적이어서 모델의 정확도를 높이는 데 어려움이 있었기 때문에 향후 더 많은 데이터를 수집하고 체계적인 라벨링 작업을 진행하여, 모델의 파인튜닝을 통해 정확도를 향상시킬 계획이다. 특히 다양한 사고 유형과 차량 모델에 대한 데이터를 확보함으로써, 모델의 일반화 성능을 높일 예정이다.

모델의 크기 최적화 작업 또한 필요하다. VGG16은 16개 레이어로 구성된 복잡한 구조를 가지고 있어 메모리 사용량이 많아지는 문제가 발생하였다. 이를 해결하기 위해 본 프로젝트와 관련된 데이터셋을 활용하여 모델을 재학습하는 방법과 함께 경량화 모델인 MobileNet 및 SqueezeNet을 도입할 계획이다. 이러한 경량화 접근법을 통해 메모리 사용량을 줄이고, 실시간 처리에 적합한 효율적인 AI 시스템을 구현할 수 있을 것으로 기대된다.

6. 별첨

6.1 참고 문헌

6.1.1 Frontend

- 테지마 타쿠야, 요시다 타케토 [타입스크립트, 리액트, Next.js로 배우는 실전 웹 애플리케이션 개발]
- NEXT.JS 공식문서

6.1.2 Backend

- Spring Boot 공식문서

6.1.3 AI

- Python 공식문서

6.2 팀원 소개

- 여승철 (팀장)
역할: 프론트엔드 개발
- 김찬일 (팀원)
역할: AI 파손 판정 모델 개발, AI 웹 서버 구축
- 김태현 (팀원)
역할: 기획 담당, 화면 프로토타이핑
- 정석현 (팀원)
역할: 백엔드 개발, 이미지 S3 연동 및 커뮤니티 개발
- 조영무 (팀원)
역할: 백엔드 개발, 인증 및 입찰 시스템 구현

6.3 소스 코드

팀프로젝트 - <https://github.com/carumuch>

AI - <https://github.com/carumuch/carumuch-ai>

백엔드 - <https://github.com/carumuch/carumuch-backend>

프론트엔드 - <https://github.com/carumuch/carumuch-frontend>

6.4 시연영상 링크

https://youtu.be/xeeVluCMCA?si=b73XpNkdv0_Vsmey

6.5 발표 자료



CONTENTS	01	02	03
	팀원 소개	프로젝트 추진 일정	연구 목적 및 필요성
	04	05	06
	관련 기술 소개	아키텍처 구상도	서비스 구성
	07	08	09
	시연영상	결론 및 기대효과	QnA

01 팀원소개



여승철 조장

- 프론트엔드 개발



김찬일 조원

- AI 파손 판정 모델 개발
- AI 웹 서버 구축



김태현 조원

- 기획 담당
- 화면프로토타이핑



정석현 조원

- 백엔드 개발
- 이미지 S3 연동
- 커뮤니티 개발



조영무 조원

- 백엔드 개발
- 유저 인증 구현
- 입찰 시스템 구현

카우머치 CAR U MUCH?

NO.02

프로젝트 추진 일정

분야	과업	개발 기간							
		3월	4월	5월	6월	7월	8월	9월	10월
기획	자료 조사								
	개념 증명								
	기능 명세서 작성								
개발	DB, API 설계 및 구현								
	디자인 시스템 구축								
	Image Segmentation								
	파손도, 손해사정 모델 구현								
	로직 연동								
배포	배포 환경 선형								
	CI/CD 구현								

연구 목적 및 필요성

[인공지능을 통한 차량 파손 분석 및 견적 입찰 시스템]

복잡한 사고 처리 과정

차량 사고 발생 후 전문가의 현장 방문, 손상 평가, 보험사와의 협의 등 여러 단계를 지나야 하며, 이는 차주에게 큰 부담을 주지만, 이를 해결하기 위한 방안이 마땅히 없다.

AI 기반 손상 판별

AI 기술을 활용하여 대량의 데이터를 학습함으로써 미세한 외관 손상까지 판별할 수 있으며, 기존의 일방적인 평가 방식에 대한 주관성을 완화할 수 있는 보조 지표가 될 수 있다.

비용 및 시간 절약

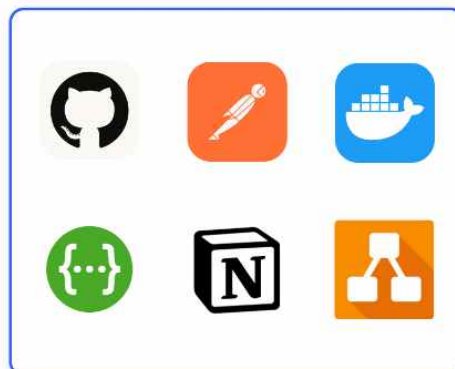
사용자는 파손 사진을 업로드 하는 간단한 방식으로 AI 기반 손상 분석과 견적 산출을 빠르게 받아 볼 수 있다. 또한 입찰 및 견적 기능을 통해 사용자와 공업사를 신속하게 중개할 수 있다.

관련 기술 소개

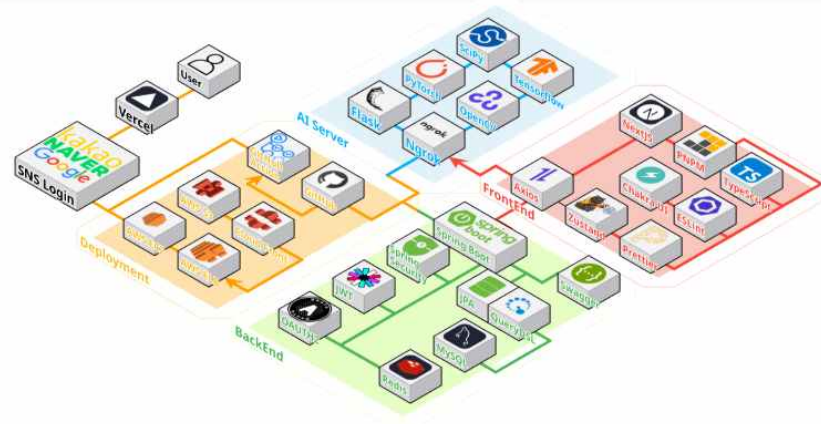
[언어 및 프레임워크]



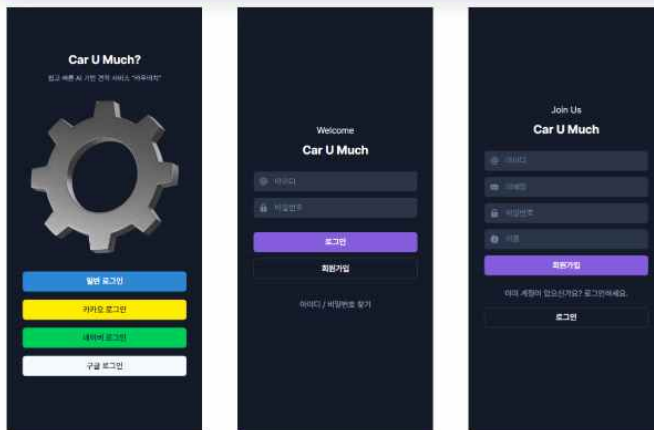
[도구, 인프라]



아키텍처 구상도

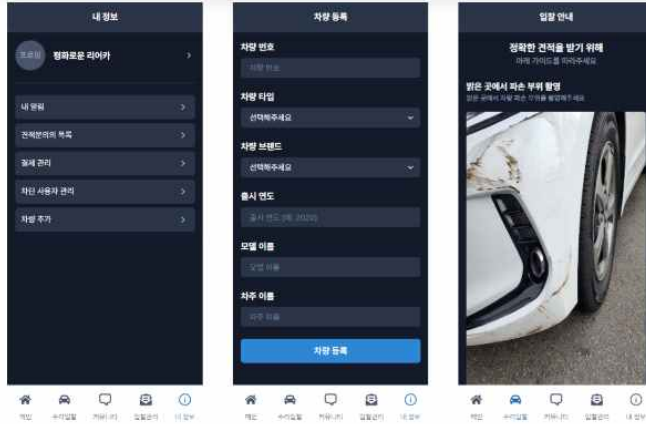


서비스 구성



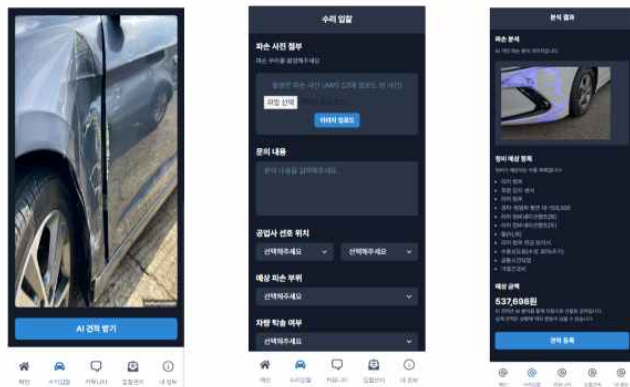
- SNS 로그인 지원
네이버, 카카오, 구글 지원
- 프론트엔드에서 정규표현식을 이용해서 1, 차로 아이디 비밀번호 규칙을 검증하고, 2차로 백엔드에서 값을 검증.
- 서버가 인증번호를 생성하고 Redis TTL 기능을 통해 120초 동안 저장하여 사용자 이메일로 인증 후 JWT 발급

서비스 구성



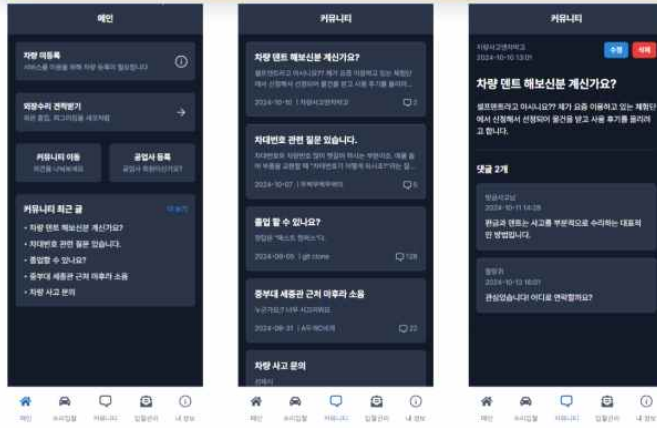
- 공업사 회원이 아닌 사용자는 시견적을 받기 위해 내 정보에서 차량 추가를 해야 한다.
- 수리입찰 가이드를 따라 공업사 회원이 아닌 사용자는 본인의 차량을 등록한 후 견적 시받기를 터치하면 수리 입찰 화면으로 이동하게 된다.

서비스 구성



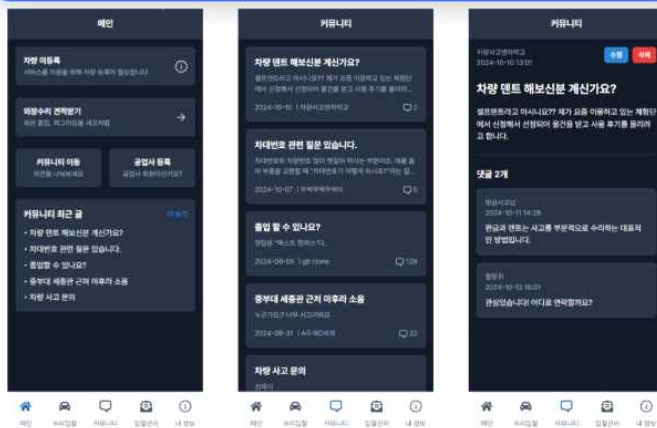
- 사용자는 차량 사고 사진 및 문의내용, 공업사 선호 위치, 예상 파손 부위 등을 체크 후 견적을 접수
- 클라이언트가 이미지를 첨부하면, 이미지 파일 이름에 UUID를 추가하여 중복을 방지하고, 백엔드 서버를 통해 S3에 이미지를 업로드한 후, 해당 이미지의 S3 URL을 반환하다

서비스 구성



- 커뮤니티는 일반적인 게시판 기능과 같이 사용자들이 의견을 나눌 수 있는 기능이다.
- 커뮤니티 글에는 사용자들이 댓글을 작성할 수 있으며, 수정을 터치시 본인이 작성한 글을 자유롭게 수정·삭제 할 수 있다.

서비스 구성



- 커뮤니티는 일반적인 게시판 기능과 같이 사용자들이 의견을 나눌 수 있는 기능이다.
- 커뮤니티 글에는 사용자들이 댓글을 작성할 수 있으며, 수정을 터치시 본인이 작성한 글을 자유롭게 수정·삭제 할 수 있다.

시연영상



결론 및 기대효과

[인공지능을 통한 차량 파손 분석 및 견적 입찰 시스템]

사고 처리 절차의 간소화

차주가 사고 차량의 사진을 업로드하면 인공지능이 자동으로 손상을 분석하고 예상 수리 비용을 신속하게 제공하여 사고 처리 절차를 간소화한다.

사용자간 정보 공유 강화

커뮤니티 기능을 통해 고객 간의 리뷰 및 정보 공유가 가능해져, 보다 객관적이고 합리적인 결정이 이루어질 수 있다.

투명한 사고 처리 방식

시스템을 통해 보험사와 공업사 간의 협업을 강화하고, 고객의 만족도를 높이며 사고 처리 과정의 투명성을 확보할 수 있다.

NO.09

Thank you

QnA