

# 기술 설계서

## 중고차 수출 중개 플랫폼

국내 차주와 해외 바이어를 연결하는 B2B, B2C 중고차 수출 중개 플랫폼의 해외 바이어용 웹 서비스입니다.

### 1. 회원가입

해외 바이어는 플랫폼과의 신뢰(Trust) 확보가 가장 중요하기 때문에,

가입 단계에서 가능한 많은 정보를 수집하여 바이어의 실존성·신뢰도·진정성을 확인하는 구조로 설계합니다.

#### 회원가입 플로우

##### 1. 가입 폼 작성

- 일반 가입보다 더 많은 항목을 요구
- “신뢰 기반 플랫폼”이라는 메시지 노출

##### 2. 이메일 실시간 중복체크

- 기존 계정 확인

##### 3. 여권번호/사업자번호 유효성 패턴 체크

- 국가별 패턴 검사(완벽 검증은 불가해도 신뢰도 ↑)

##### 4. 가입 제출

##### 5. 자동 승인 (ROLE\_BUYER 부여)

- 단, 운영팀은 관리 화면에서 바이어 정보를 다시 확인할 수 있음
- 추가 검증 플래그(verified)가 별도로 존재

### 2. 메인화면 - AI 검색 + 경매 + 차량 탐색

#### AI 기반 차량 조건 파싱

##### 1. 사용자가 자연어로 조건 입력

("Budget 10k~13k, mostly Toyota SUV for Kenya shipment")

2. 백엔드가 LLM에게 템플릿 기반 파싱 요청
  3. LLM이 구조화된 필드 JSON 반환
    - brand / model / year / mileage
    - fuel\_type
    - price\_range
    - country
  4. 누락 필드는 NULL → 전체 범위 자동 확장
  5. Query Builder를 통해 최적화된 조회 실행
- 

## 경매 시스템

### Auction (입찰 방식)

- 시작가부터 여러 바이어가 입찰
- 종료 시 최고가가 낙찰
- 입찰 경쟁이 많기 때문에 낙관적 락(optimistic)으로 동시성 보장

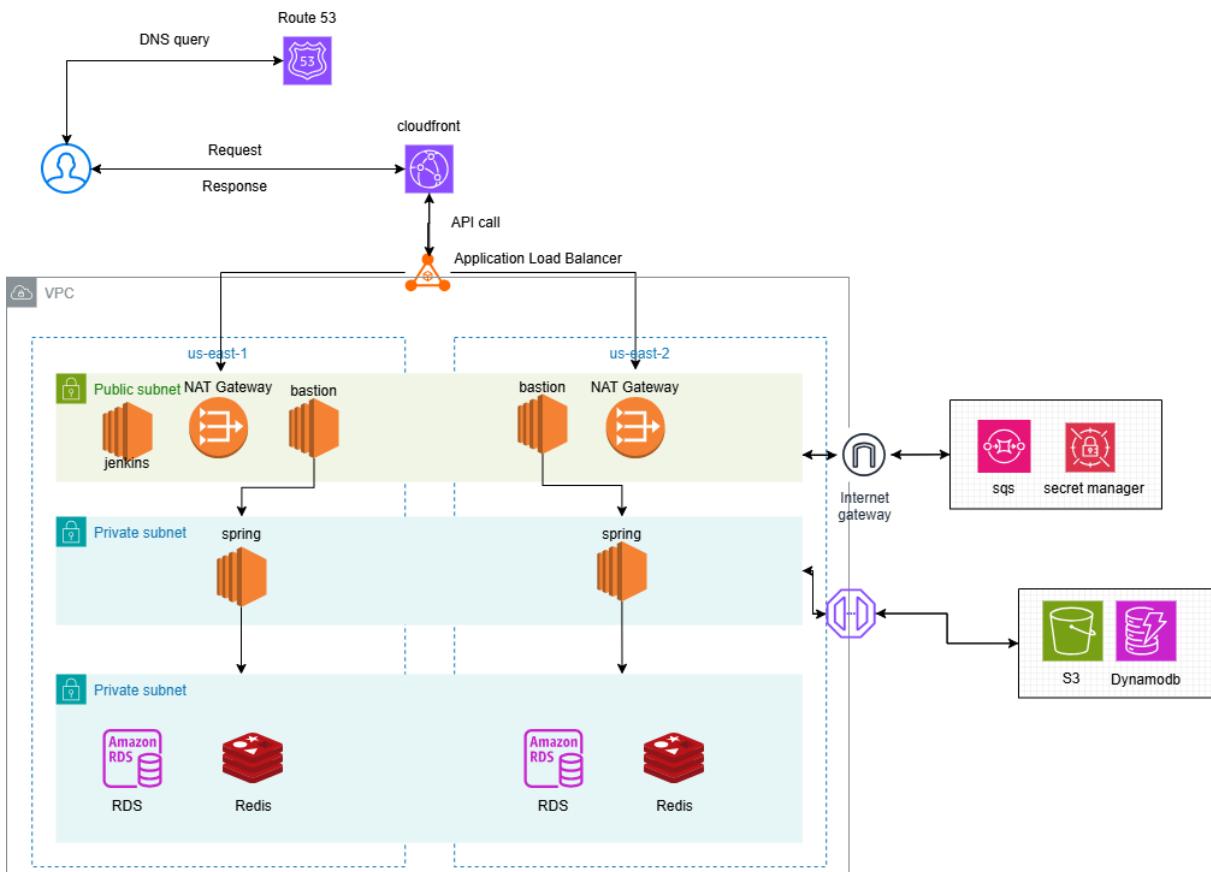
### Buy Now (즉시구매)

- buy\_now\_price 이상을 제시하면 즉시 낙찰
  - 첫 번째 성공 요청만 인정 (원자성 보장)
- 

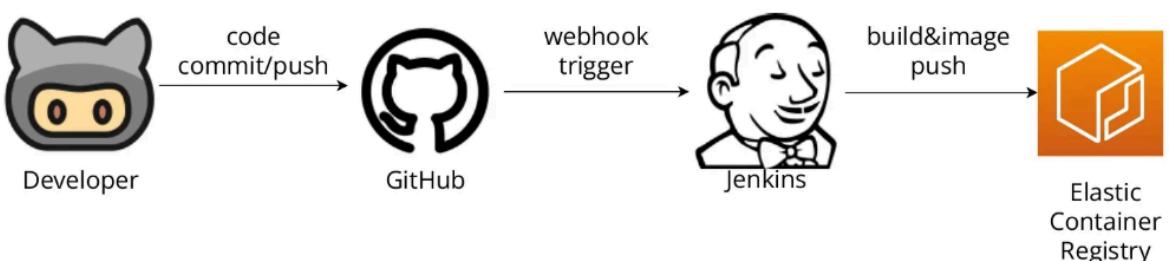
## 전체 차량 조회 성능 최적화

- 주요 컬럼 인덱싱(price, mileage, year, country, fuel\_type)
  - 정렬 기준 인덱스 분리
  - 무한 스크롤 형식 적용
  - 검색 폭 확장 시 조건이 많아도 빠른 조회 가능
-

# 인프라 설계



## CI/CD



# 인프라 설계 플로우

## ▼ vpc 생성

- public
  - ALB : alb에서만 ec2 접근가능하게 하기(보안)
  - bastion : ec2(spring 서버)에 접속하기 위함
  - jenkins : jenkins 서버구축
- private
  - RDS, S3, ec2(spring 서버)

## ▼ subnet 생성

- 두 region에 public, private 하나씩 만들기
- region a에 전부다 기능 구현
- region b는 고가용성(장애발생시 사용)
- public은 퍼블릭 주소 자동 할당 활성화

## ▼ IGW, NAT gateway, RDS

- internet gateway
  - 외부에서 public subnet 찾아올 때 필요
  - igw는 vpc에 연결을 해주는거고 routing table을 통해 인터넷을 통과할 subnet 선택
- nat gateway
  - nat gateway는 프라이빗 서브넷이 외부와 통신하기 위해 필요하다. 보통 퍼블릭에 만들어 프라이빗에서는 라우팅 테이블로 nat gateway로 통신이 되어야 한다. 따라서 nat gateway 설정에서 탄력적 ip를 받아 ip를 고정시켜야한다.
  - 연결유형은 public
  - routing table은 private subnet에 연결해주면 된다.
- rds subnet 그룹
  - db를 위한 subnet 그룹 생성

## ▼ security group 설정

이름	대상	허용 인바운드	설명
cariv-sg-alb	ALB	80, 443 (0.0.0.0/0)	외부에서 API 호출 용
cariv-sg-app	EC2(Spring Boot)	8080 (from sg-alb), 22 (from sg-bastion)	ALB/SSH 접근 허용
cariv-sg-bastion	Bastion EC2	22 (My IP)	관리용 SSH
cariv-sg-rds	RDS	3306 (from sg-app)	App 서버만 DB 접근 허용
cariv-sg-jenkins	jenkins EC2	22(My IP), 8080(My IP)	jenkins 용

#### ▼ ec2 생성

- spring ec2
  - private
  - spring 서버 돌릴 예정
  - rds, s3에 접근 가능해야 함
  - alb에서 접근
- bastion ec2
  - public
  - 관리용
  - spring ec2에 접속하기 위해서..
  - ssh
  - EIP 해주기

#### ▼ bastion 서버로 접속하고 spring 서버 접근하기

- sg 먼저 bastion 서버만으로 해주기(sg-app)
- 그냥 접속하면 pem키 없으니까 pem키 bastion 서버에 넣어주기

#### ▼ RDS

- public 접근 안되게 만들기

## ▼ route53

- https를 alb로 연결하려면 도메인이 필요(우선 가비아에서 구매- mumuri.shop)
- route53에서 name server 정보 4개 받아서 가비아 ns에 수정

## ▼ alb

- 인터넷 경계로 만들기
- https에 대한 리스너가 필요
  - 대상 그룹 443으로 만들어주기
  - 보안 정책에서 acm이 필요한데 이거 발급받으려면 도메인 필요(가비아에서 구매)

## ▼ Jenkins

- ec2에 jenkins 설치하기

```
sudo apt update
sudo apt install openjdk-17-jdk -y
sudo mkdir -p /usr/share/keyrings
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | \
sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] htt
ps://pkg.jenkins.io/debian-stable binary/" | \
sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt update
sudo apt install jenkins -y
//nameserver를 8.8.8.8로 바꾸기(plugin 설치가 잘안될때..)
sudo sed -i '1inameserver 8.8.8.8' /etc/resolv.conf
```

- 접속해서 plugin 설치

```
Git plugin
GitHub Integration plugin
Pipeline
SSH Agent
Credentials Binding
Gradle
Docker Pipeline
```

AWS Credentials

Amazon ECR

Amazon ECS

Blue Ocean

AnsiColor

Pipeline Utility Steps

- jenkins credential 추가하기
- github에는 webhook 설정해주기

▼ ecr

- iam user에 jenkins 전용 유저 만들어 주기
- access key랑 secret 키 받고 ecr 관련 권한 설정해주기
- jenkins → aws credential에 등록

# 플랫폼 기능상세

---

## 회원가입(Registration) – 개인 / 사업자 분기

회원가입은 사업자와 개인으로 분리된 플로우.

---

### 공통 단계

- 사업자/개인 선택
  - 구매 목적 선택
    - Personal Use
    - Corporate / Company
    - Retail Resale
    - Wholesale
    - Rental / Leasing
    - Taxi / Ride Service
    - Spare parts / Salvage
  - 연간 구매 수량 (1~100대)
  - 이메일 입력
  - 비밀번호 입력
- 

### ① 글로벌 바이어(개인)

#### 1) 기본 정보

- 영문 이름(여권 일치)
- 생년월일
- 국적
- BL 수령용 주소지

#### 2) 여권 정보 입력

- 여권번호

- 만료일

### 3) 연락처 입력

- 국가 선택
- 전화번호
- WhatsApp 연동, Telegram 연동

### 4) 동의

- 이용약관
- 개인정보 처리방침
- 마케팅 동의

---

## ② 글로벌 바이어(사업자)

### 1) 회사 정보

- 회사명(영문)
- 사업자등록번호
- 소재 국가
- BL 수령용 주소

### 2) 담당자 정보

- 담당자 영문 명
- 직책
- 전화번호
- 담당자 이메일

### 3) 회사 유효성 검증

- 사업자등록증 업로드
- 홈페이지(옵션)

### 4) 연락처

- 회사 공식 전화번호

- WhatsApp, Telegram 연동

## 5) 동의

- 이용약관
- 개인정보처리방침
- 마케팅 동의

## 홈 화면 구성요소

- 프롬프트 박스 (AI Search Box)
- Quick Search(퀵서치)
- Partners(파트너 로고 단)
- All Cars(전체 차량 리스트)

### ① 프롬프트 박스

사용자 입력 예시:

"Show me low-mileage SUVs under \$7,000."

### AI가 파싱해야 하는 필드

필드	설명
manufacturer	제조회사
model	모델명
year	연식
mileage	키로수
fuel_type	연료 타입
country	나라
price	가격

NULL이면 → SELECT \* 처리

즉,

- “키로수 안 말함” → WHERE mileage LIKE ‘%’
- “연료타입 말 안함” → fuel\_type 조건 제외

## 프롬프트 파싱 구조

User Query → Prompt Template → LLM → JSON → Search Query Builder → 결과 리턴

## ② Quick Search

사용자가 조건을 직접 선택해서 검색하도록 하는 필터형 UI

### Quick Search 필터

- 가격
- 키로수
- 연식
- 차종
- 연료 타입
- 브랜드
- 모델

| Quick Search는 “빠른 탐색”의 목적이라 필터는 최소한으로 구성.

## ③ Partners 영역

- 중고차 업계 신뢰 확보용
- 파트너 로고 4~5개
- 좌우 슬라이드 가능
- 클릭 시 상세 파트너 페이지 이동 가능

## ④ All Cars (전체 차량 리스트)

표시 요소:

- 차량 이미지
- 제조회사
- 모델명
- 연식
- 키로수
- 연료타입
- 나라
- 가격

### UI 특징

- 한 줄에 4개
  - Infinite Scroll(무한스크롤)
  - 판매방식(경매 or 즉시구매)에 따라 라벨링 다르게 표기
- 

## 차량 검색

홈의 3개 경로 중 하나로 진입 가능:

- 좌측 상단 “Browse Cars” 메뉴
  - 프롬프트 검색 결과 리스트
  - Quick Search 결과 리스트
- 

## Browse Cars 페이지 구성

### ① 상단 프롬프트 박스

홈과 동일하게 자연어 검색 가능.

---

### ② 차량 리스트

데이터 구성:

- 이미지
- 제조사
- 모델명

- 연식
- 키로수
- 연료
- 국가
- 가격
- 판매 방식(Auction / Buy Now)

#### 경매 방식인 경우

- 경매 시작가(start\_price)
- 즉시 구매가(buy\_now\_price)
- 남은 시간(time\_left) 표시

#### 즉시구매 방식

- buy\_now\_price만 표시
- 

### ③ 고정 사이드바 (좌측 or 우측)

#### 환율 정보

- 국가 선택 →
  - Google Exchange API(미정) 호출
  - 환율 표시
  - 가격 자동 변환(예: USD ↔ KES ↔ AED)
- 

### ④ 필터 기능

홈 Quick Search보다 훨씬 상세한 필터

#### 상세 필터

- 차종
- 제조사
- 모델
- 연식(year range)
- 키로수(mileage range)

- 연료 타입
- 나라(country)
- 가격(price range)
- 판매방식(Auction / Buy Now)
- 내부 색상
- 외부 색상
- 변속기
- 옵션(썬루프, 내비게이션 등)

### 즉시 반응(Instant Filtering)

- 선택하는 즉시 리스트 재조회
  - 선택된 필터는 색 강조
  - 이미 적용된 조건은 하이라이트
-

## 문서 자동화 플랫폼(수출 서류 자동화 SaaS)

한국에서 차량을 수출할 때 발생하는 말소증 / 수출신고필증 / BL / Invoice / Packing List / 차량 사진 / 성적서 등 수많은 파일들이 흩어져 있고 사람별로 전달 체계가 달라서 누락, 오타, 중복전달이 빈번합니다. 이 플랫폼은 각 차량 단위로 모든 서류를 자동 정리하고, 필요한 파일을 자동 생성·공유하여 수출업자·관세사·포워더·바이어가 같은 문서 집합을 한 번에 볼 수 있게 만드는 시스템입니다.

---

### 플로우

1. 사용자가 파일 업로드
2. 시스템이 자동으로 문서 종류 분류
3. OCR로 주요 차량 정보를 읽어 필드 테이블 생성
4. 필드 기반으로 Invoice, Packing List 자동 생성
5. 차량 단위 문서 패키지 구성
6. 수출업자·관세사·포워더와 공유
7. 문서 누락 또는 수정 시 바로 업데이트

### 서류 범위

처리 대상은 다음에 한정:

- 말소증
  - 수출신고필증
  - BL(선적 관련 문서)
  - 차량 사진
  - 인보이스 생성에 필요한 기본 차량정보
  - 패킹리스트 생성용 필드
  - 관세/포워더가 추가 업로드하는 파일들
- 

### 주요 기능

#### (1) 파일 업로드

- 단일·다중 파일 업로드 가능
- 드래그&드롭 영역 필요
- 업로드 후 즉시 "차량 단위"로 묶기

#### 화면 요소

- 업로드 박스
  - 업로드한 파일 목록
  - 파일 타입 자동 라벨링
- 

## (2) 문서 종류 자동 분류

업로드된 파일은 아래 카테고리로 자동 라벨링:

- 말소증
- 수출신고필증
- BL
- Invoice 관련 자료
- Packing List 관련 자료
- 차량 사진
- 기타 문서

#### 요구사항

- 파일 카드 컴포넌트
  - 자동/수동 라벨 변경 가능
  - 미분류된 파일에 경고 띄움
- 

## (3) OCR 기반 차량정보 자동 추출

처리 로직: 한국 문서만을 대상으로 OCR → 필드 테이블 생성.

#### 필드 예시

- 차대번호(VIN)
- 차량번호
- 차종

- 연식
- 신고번호
- 말소일자
- BL 번호 등

#### 요구사항

- 필드 테이블 UI (엑셀처럼)
- 잘못된 값은 직접 수정 가능
- 실시간 자동 저장

---

## (4) 자동 문서 생성 (PDF/Excel)

OCR 결과 기반으로 자동 생성:

- Invoice PDF
- Packing List PDF
- Excel/CSV Export
- 업체별 템플릿 커스텀 가능

#### 요구사항

- “문서 생성” 버튼
- 생성된 문서 미리보기 Modal
- 다운로드 버튼

---

#### 요구사항

- 정렬된 패키지 리스트
- 패키지별 상태 표시(서류 누락 여부)

---

## (6) 협업 공유 (수출업자 ↔ 관세사 ↔ 포워더)

- 링크 공유 기능
- 권한: 읽기 / 수정
- 누락 파일 요청 메시지 자동 생성(서버에서 템플릿 제공)

# 인력 구성 및 운영 개요

## 개발 방식 (Agile Sprint 기반 페이지 단위 개발)

완전한 와이어프레임/디자인이 100% 완성된 후 개발하는 방식이 아니라, 페이지 단위로 기획 → 와이어 → 디자인 → 개발이 동시에 굴러가는 Agile 방식으로 진행할 예정입니다.

“필요한 화면의 기획이 나오면 즉시 디자인 → 즉시 개발 → 즉시 테스트” 한 페이지 단위의 mini sprint 구조

## 인력 구성

### 플랫폼 개발팀 – 총 4명

역할	인원	주요 업무
<b>Backend Developer</b>	2명	Spring 기반 API / 경매 시스템 / AI 검색 / DB 설계
<b>Frontend Developer</b>	1명	해외 바이어 웹 UI, 반응형 화면, 검색/경매 페이지
<b>Designer</b>	1명	웹 UX/UI, 가입 화면, 메인 구조, 브랜드 가이드

### 문서 자동화 프로그램 개발팀 – 총 3명

역할	인원	주요 업무
<b>Backend Developer</b>	1명	OCR Pipeline, 문서 Parsing, 변환 엔진(PDF/Excel)
<b>Frontend Developer</b>	1명	파일 업로드 UI, Dashboard, DMS 화면
<b>Designer</b>	1명	문서 분석 UI/흐름, 편집기 UX, 관리자 대시보드

# 전체 일정

## 1) 문서 자동화 프로그램 개발 일정 (총 7~8주)

### 기획 & 분석 (2~3주)

- 수출·서류 플로우 현장 학습
- Deregistration, Export Cert, Invoice 등 문서 구조 분석
- OCR 필드 매팅 정의
- Template 도식화
- API 스펙 결정
- 최종 UI Wireframe 도출

### 집중 개발 (4주)

#### Backend

- OCR Pipeline 구현
- Upstage/Custom API 연동
- Parsing Layer + Validation Layer
- 문서 변환(Invoice/PDF/Excel) 엔진
- S3 저장 / 권한 관리 / Versioning
- API 배포

#### Frontend

- 파일 업로드 화면
- Dashboard / DMS UI
- 출력물 다운로드
- 문서 상세 비교 화면

#### Designer

- UX 개선
- 관리자 View 디자인
- 차량별 문서 묶음 디자인

# 플랫폼(중고차 수출) 개발 일정 – 3월 MVP 목표

## 플랫폼 핵심 기능 개발

### Sprint 1 (1주)

- 회원가입 페이지
- 이메일/여권번호 검증 API
- 신뢰 기반 입력폼 완성

### Sprint 2 (1주)

- 로그인 + 권한 시스템
- 바이어 대시보드 기본 틀

### Sprint 3 (1~1.5주)

- AI 검색 입력창
- LLM 파싱 엔진
- Query Builder 검색 API

### Sprint 4 (1주)

- 검색 결과 리스트 페이지
- 필터/정렬
- 상세 페이지 기본 구조

### Sprint 5 (1~1.5주)

- 경매 생성/상세 페이지
- 입찰 API
- Buy Now 로직
- 동시성 처리

### Sprint 6 (1주)

- 경매 Countdown + UI 연동
- 카운트다운 로직 강화

## Sprint 7 (0.5주)

- 전체 성능 튜닝
- 데이터 세팅
- UI 폴리싱

이후 기능 통합테스트 및 디자인 정리, 보수 작업

---

## 협업 방식

### ✓ 회의 / 스크럼

- 주 1회 온라인 회의(Discord)
- 주 1회 스크럼(Discord)
- 필요 시 격주 또는 월 1회 대면 회의
- 회의 목적
  - 주간 리뷰
  - 기능 우선순위 조정
  - 문제/리스크 공유
  - 스프린트 목표 확정

### ✓ 업무 관리

- Notion
  - 기획 문서
  - 스프린트 보드
  - 기능 스펙 문서
  - API 문서
- GitHub
  - Repository 분리(Platform / Document Automation)
  - Issue 기반 Task 관리

## ✓ 커뮤니케이션

- Discord 실시간 채널
- 카카오톡 서브 커뮤니케이션
- Figma로 디자인 협업 가능

## ✓ 배포

- 초기 개발 단계는 Dev 환경(Default)
- 배포 자동화는 Jenkins

# 선호 기술 스택

## 프론트엔드 개발자

- **React 기반 웹서비스 개발 경험**
- **REST API / JSON 기반 데이터 연동 경험**
  - 서버에서 내려오는 데이터 구조 이해
  - 에러 처리, 토큰 기반 인증 처리 경험
- **SaaS 형태의 대시보드·업무용 웹 개발 경험**
  - 테이블/필터/검색/정렬/폼 등 기능성 페이지 구현 경험
- **AI 디자인 툴 활용 및 디자이너와의 커뮤니케이션이 능숙한 분**
- **협업 경험**
  - Git 기반 협업, PR 리뷰 경험
  - 빠른 의사소통과 작업 공유에 익숙한 분

---

## 백엔드 개발자

- **Java / Spring Boot 기반 웹서비스 개발 경험**
  - REST API 설계·구현, 계층 구조(Service/Domain/Repository) 이해
- **MySQL 기반 DB 설계 경험**

- 정규화/인덱스/쿼리 최적화 기본 이해
  - **JWT 기반 인증/인가 구현 경험**
    - Spring Security 혹은 Custom Filter 사용해본 경험
  - **Redis 등 캐시 사용 경험**
    - 로그인 세션, 토큰 검증, 조회 성능 개선 등 활용 경험
  - **AWS 인프라 경험**
    - EC2, RDS, S3, CloudFront, Route53 중 일부라도 경험 있으면 우대
  - **Docker 기반 개발·배포 경험**
    - docker-compose 또는 간단한 배포 파이프라인 구성 경험
  - **문서 자동화·OCR·AI API 연동 경험**
  - **SaaS 백엔드 구조 이해**
    - 멀티테넌시, 권한 구조, 비즈니스 로직 분리 경험
- 

## 디자이너

- **Figma 중심의 웹/앱 UI 디자인 경험**
  - 컴포넌트 기반 설계, Auto Layout, 변수/토큰 활용 경험
- **업무용 웹서비스(SaaS) UI 제작 경험**
  - 테이블, 필터, 상세보기, 편집 화면 등 구조화 경험
- **UX 기획 경험**
  - 사용자 플로우, 와이어프레임, 시나리오 제작
- **AI 디자인 툴 활용 경험**
  - Figma AI / Framer AI / Uizard 등 빠른 시안 제작
- **개발자와 협업 경험**
  - 컴포넌트 명세 정리, 간단한 프론트 구조 이해