

ReLoop v2 — 전체 매뉴얼

적응형 복습 스케줄링 플랫폼

2026-04-12

ReLoop v2

잊기 전에 다시 만나는 학습법. 페르소나 기반 망각 곡선으로 개인화된 복습 간격을 자동으로 계산하는 적응형 학습 플랫폼.

수능, 자격증, 어학, 취미 학습 — 어떤 공부든 ReLoop이 기억을 관리합니다.

핵심 기능

기능	설명
적응형 복습 스케줄링	페르소나 + 문제 난이도에 따라 “정답률이 기준 아래로 떨어지는 시점”을 자동 계산해 복습 큐에 넣습니다
KICE 기출 문제집	2025/2026 수능 기출 (국어/수학/영어/한국사/생활과 윤리) 320문항을 PDF에서 자동 크롭한 이미지로 제공
모의고사 모드	실제 시험처럼 타이머 + OMR 답안 선택 + 자동 채점 + 학습 기록 일괄 생성
영어 듣기 오디오	영어 듣기평가 MP3를 내장 오디오 플레이어로 재생
통계 대시보드	학습 히트맵, 과목별 정답률, 숙련도 성장 곡선을 시각화
PWA 지원	모바일 홈 화면에 앱으로 설치 가능

기술 스택

영역	기술
Backend	NestJS 10 + Prisma 5 + MariaDB
Frontend	Next.js 14 (App Router) + React 18 + styled-components 6
인증	JWT (passport-jwt)
보안	helmet, @nestjs/throttler (60req/min), CORS whitelist
아이콘	Phosphor React
폰트	Pretendard + Inter + JetBrains Mono
배포	PM2 on Debian VM, nginx reverse proxy

영역 기술

PDF 처리 poppler-utils (pdftotext, pdftoppm, pdfinfo)

OCR Codex CLI Vision (codex exec -i)

프로젝트 구조

```
reloop-v2/
├── backend/                                     # NestJS API 서버
│   ├── src/
│   │   ├── auth/                             # JWT 인증 (login, register,
me) │   │   ├── dashboard/                     # 대시보드 요약 API
│   │   ├── forget/                           # 페르소나 망각 곡선 엔진
│   │   ├── me/                               # 프로필, 아바타 업로드
│   │   ├── problem-sets/                     # 문제집 CRUD + 파싱 모듈
│   │   │   └── parsing/                       # PDF 파싱 엔진 (범용, 전략 패
턴) │   │   ├── strategies/                     # kice.ts, math.ts
│   │   │   └── ocr-fallback/                 # Codex Vision OCR
│   │   ├── reviews/                         # 복습 큐 관리
│   │   ├── stats/                           # 통계 API
│   │   ├── study-logs/                       # 학습 기록 + 모의고사 일괄 제출
│   │   ├── subjects/                         # 과목 관리
│   │   └── tags/                             # 태그 관리
│   ├── prisma/
│   │   ├── schema.prisma                     # DB 스키마
│   │   ├── seed.ts                           # 데모 데이터 시드
│   │   └── migrations/                       # 마이그레이션 히스토리
│   ├── scripts/
│   │   ├── kice-import.ts                     # KICE PDF импорт CLI
│   │   └── ocr-fix.ts                         # OCR 보정 CLI
│   └── uploads/                               # 문제 이미지, 아바타 (git 제외)
├── frontend/                                  # Next.js 프론트엔드
│   ├── src/
│   │   ├── app/                               # App Router 페이지들
│   │   │   ├── dashboard/                     # 대시보드 (채움/빈 상태)
│   │   │   ├── exams/                         # 문제집 카탈로그
│   │   │   ├── study/
│   │   │   │   ├── exam/[id]/                 # 모의고사 시험 화면
│   │   │   │   └── history/                   # 학습 기록 히스토리
│   │   │   ├── review/                       # 복습 세션
│   │   │   ├── solve/[id]/                   # 개별 문제 풀기
│   │   │   ├── study-logs/[id]/              # 학습 기록 상세
│   │   │   ├── stats/                         # 통계
│   │   │   ├── subjects/                     # 과목 및 태그 관리
│   │   │   ├── profile/                      # 프로필 + 설정 통합
│   │   │   └── pricing/                      # 플랜 (공개/로그인 분기)
```

				onboarding/	# 신규 사용자 온보딩
				login/ + register/	# 인증
				privacy/ terms/ contact/	# 법적 페이지
				page.tsx	# 랜딩 페이지
				components/	
				ui/	# Button, Card, Icon, Toast,
				Select, DatePicker, ...	
				layout/	# AppShell, SideNav,
				BottomNav	
				auth/	# AuthCard (로그인/회원가입 통
				합)	
				dashboard/	# EmptyDashboard
				legal/	# LegalLayout
				lib/	# api.ts, auth.ts,
				constants.ts, exam-date.ts	
				styles/	# theme.ts (디자인 토큰, 애니메
				이션)	
				public/	# 정적 파일, PWA manifest, 아
				이콘	
				data/kice/	# KICE 원본 PDF (git 제외, 저
				작권)	
				2025/{국어,수학,영어,...}/	
				2026/{국어,수학,영어,...}/	
				docs/	# 상세 매뉴얼
				algorithm.md	# 망각 곡선 알고리즘 상세
				architecture.md	# 시스템 아키텍처
				user-guide.md	# 사용자 가이드
				admin-guide.md	# 운영/배포 가이드
				pdf-pipeline.md	# PDF 파싱 파이프라인
				Plans.md	# 개발 계획 + 진행 상황

알고리즘 요약

상세 내용: <docs/algorithm.md>

ReLoop은 **FSRS** 대신 자체 페르소나 기반 망각 곡선을 사용합니다.

핵심 수식

실력 감쇄: $S(t) = S_0 \times \exp(-\lambda \times t)$
 정답 확률: $P = \text{sigmoid}(k \times (S(t) - D))$

- **S_0** (0~1): 마지막 학습 시점의 실력. 맞으면 올라가고 틀리면 떨어짐
- **λ (lambda)**: 페르소나별 망각 속도. senior(0.1)은 느리게, crammer(0.6)은 빠르게 잊음
- **D** : 문제 난이도 (0~1)
- **k** : sigmoid 기울기 (기본 4.0)

복습 시점 결정

“정답 확률 P가 사용자의 기준(P_threshold) 아래로 떨어지는 시점”을 다음 복습일로 잡습니다.

복습 민감도	P_threshold 의미	
엄격 (strict)	0.7	70% 이하로 떨어지면 복습
균형 (moderate)	0.5	50% 이하
여유 (relaxed)	0.35	35% 이하

페르소나

페르소나	λ (망각속도)	복습 주기 경향
Senior	0.1	7~14일 (느린 망각)
Mid	0.2	2~5일
Junior	0.4	1~2일 (빠른 망각)
Crammer	0.6	매일~격일

So 업데이트 규칙

결과 새 **So**

정답 $S_o \times 0.5 + 0.6$

부분 $S_o \times 0.7 + 0.3$

오답 $S_o \times 0.5$

빠른 시작

필수 조건

- Node.js 20+
- pnpm 9+
- MariaDB/MySQL 8+
- poppler-utils (pdftotext, pdftoppm, pdfinfo)

설치 및 실행

1. 클론

```
git clone ssh://git@your-gitea:2222/hanarang/reloop-v2.git
cd reloop-v2
```

2. 백엔드

```
cd backend
pnpm install
cp .env.example .env          # DATABASE_URL, JWT_SECRET 편집
```

```
pnpm prisma migrate dev
pnpm prisma:seed          # 데모 계정 생성
pnpm dev                  # http://localhost:3001
```

3. 프론트엔드 (새 터미널)

```
cd frontend
pnpm install
cp .env.example .env.local # NEXT_PUBLIC_API_URL 편집
pnpm dev                  # http://localhost:3000
```

4. (선택) KICE 기출 импорт # data/kice/ 에 PDF 배치 후:

```
cd backend
pnpm cli:kice-import --year=2026
```

데모 계정

이메일	비밀번호
demo@reloop.local	demo1234

환경 변수

Backend (backend/.env)

변수	필수	기본값	설명
DATABASE_URL	O	—	mysql://user:pass@host:3306/reloop_v2
JWT_SECRET	O	—	32자 이상 랜덤 문자열
JWT_EXPIRES_IN	—	7d	토큰 만료 기간
CORS_ORIGINS	—	http://localhost:3000	웹표 구분 허용 도메인
PORT	—	3001	API 서버 포트

Frontend (frontend/.env.local)

변수	필수	기본값
NEXT_PUBLIC_API_URL	—	http://localhost:3001/api

배포

상세 내용: [docs/admin-guide.md](#)

서버에서

```
cd /path/to/reloop-v2
```

백엔드

```
cd backend && pnpm install && pnpm prisma:deploy && pnpm build  
pm2 restart reloop-api
```

프론트엔드

```
cd frontend && pnpm install && pnpm build  
cp -r .next/static .next/standalone/.next/  
cp -r public .next/standalone/  
pm2 restart reloop-web
```

API 요약

상세 내용: [docs/architecture.md](https://docs.reloop.dev/docs/architecture.md)

Method	Endpoint	설명
POST	/api/auth/register	회원가입
POST	/api/auth/login	로그인 (JWT 발급)
GET	/api/auth/me	내 정보
PATCH	/api/me/profile	프로필 수정
POST	/api/me/avatar	아바타 업로드
GET	/api/dashboard/summary	대시보드 요약
GET	/api/problem-sets	문제집 목록
GET	/api/problem-sets/:id	문제집 상세 (문항 + 지문)
GET	/api/study-logs	학습 기록 목록
POST	/api/study-logs	학습 기록 생성
GET	/api/study-logs/:id	학습 기록 상세
PATCH	/api/study-logs/:id	학습 기록 수정
POST	/api/study-logs/from-problem-set	모의고사 일괄 제출
GET	/api/reviews/queue	오늘의 복습 큐
POST	/api/reviews/:id/submit	복습 결과 제출
GET	/api/subjects	과목 목록
GET	/api/stats/subjects	과목별 통계
GET	/api/stats/mastery-path	숙련도 성장 곡선

라이선스

UNLICENSED (비공개)

ReLoop 알고리즘 상세

개요

ReLoop은 에빙하우스 망각 곡선(Ebbinghaus forgetting curve)에 “페르소나”라는 개인화 레이어를 얹은 자체 복습 스케줄링 알고리즘을 사용합니다. FSRS(Free Spaced Repetition Scheduler)나 Anki의 SM-2와는 다른 독자적 모델입니다.

핵심 아이디어: 사람마다 잊는 속도가 다르고, 같은 사람도 쉬운 문제와 어려운 문제를 다른 속도로 잊는다. ReLoop은 이 두 축(페르소나 × 난이도)을 수학적으로 모델링합니다.

수학적 모델

1. 실력 감쇄 함수 (Skill Decay)

$$S(t) = S_0 \times \exp(-\lambda \times \Delta t)$$

- **S₀**: 마지막 학습/복습 직후의 실력 수준 (0~1 사이 실수)
- **λ (lambda)**: 페르소나별 망각 속도 계수 (단위: 1/일)
- **Δt**: 마지막 학습으로부터 경과한 일수
- **S(t)**: 현재 시점의 실력 수준

시간이 지날수록 S(t)는 지수적으로 감소합니다. λ가 클수록 빠르게 잊습니다.

2. 정답 확률 함수 (Recall Probability)

$$P(\text{correct}) = \sigma(k \times (S(t) - D))$$

여기서 σ는 시그모이드 함수:

$$\sigma(x) = 1 / (1 + \exp(-x))$$

- **k**: 시그모이드 기울기 상수 (기본값 4.0). 클수록 실력과 난이도의 차이에 민감하게 반응
- **D**: 문제 난이도 (0~1). 기출 문제의 경우 1 - 공개된정답률로 계산
- **P**: 이 문제를 지금 풀면 맞힐 확률 (0~1)

직관적으로: 실력 **S(t)**가 난이도 **D**보다 높으면 맞힐 확률이 높고, 낮으면 떨어집니다.

3. 복습 시점 결정 (Scheduling)

사용자의 복습 민감도(intensity)에 따라 P_threshold가 정해지고, **P**가 이 기준 아래로 떨어지는 시점을 다음 복습일로 잡습니다.

방정식을 Δt에 대해 풀면:

$$\Delta t = -(1/\lambda) \times \ln((D + \text{logit}(P_threshold)/k) / S_0)$$

여기서 $\text{logit}(p) = \ln(p / (1-p))$ 입니다.

4. 경계 조건

상황	처리
계산된 $\Delta t < 0$ (이미 기준 이하)	1시간 후 복습
target 값 ≤ 0 (매우 쉬운 문제)	60일 후 (최대 간격)
target $> S_0$ (실력이 기준에 한참 못 미침)	1시간 후 복습
$\Delta t > 60$ 일	60일로 캡

페르소나 시스템

사용자는 4가지 페르소나 중 하나를 선택합니다. 각 페르소나는 서로 다른 망각 속도(λ)를 가집니다.

페르소나별 λ 값

페르소나	λ 의미	적합한 학습자
Senior (숙련자)	0.1 천천히 잊음	기초가 탄탄한 학습자. 한 번 배우면 오래 기억
Mid (중급)	0.2 보통 속도	대부분의 학습자에게 추천 (기본값)
Junior (초보)	0.4 빠르게 잊음	새로운 분야를 시작한 학습자
Crammer (단기집중)	0.6 매우 빠르게 잊음	시험 직전 단기 반복 학습

실제 복습 주기 예시 (난이도 0.5, **moderate** 민감도)

페르소나	정답 후 S_0	다음 복습까지
Senior	0.75	~4일
Mid	0.75	~2일
Junior	0.75	~1일
Crammer	0.75	~12시간

복습 민감도 (Review Intensity)

설정	$P_threshold$	의미
엄격 (strict)	0.70	정답 확률이 70% 아래로 떨어지면 바로 복습. 가장 자주 복습
균형 (moderate)	0.50	50% 기준. 추천 기본값
여유 (relaxed)	0.35	35%까지 허용. 최소한의 복습으로 유지

S₀ 업데이트 규칙

학습/복습 결과에 따라 실력 값 S₀가 업데이트됩니다:

결과	수식	예시 (이전 S₀=0.5)
정답	$\text{new_}S_0 = \text{old_}S_0 \times 0.5 + 0.6$	0.85
부분 정답	$\text{new_}S_0 = \text{old_}S_0 \times 0.7 + 0.3$	0.65
오답	$\text{new_}S_0 = \text{old_}S_0 \times 0.5$	0.25

- 정답을 맞히면 S₀가 크게 올라가고, 오답이면 절반으로 떨어집니다.
- S₀는 항상 0~1 범위로 클램핑됩니다.
- 첫 학습 시 S₀ 초기값은 문제의 baseCorrectRate (공개 정답률)이 있으면 그 값을 기반으로, 없으면 0.3을 사용합니다.

Anki / FSRS와의 차이

	ReLoop	Anki (SM-2)	FSRS
개인화 축	페르소나 (4종)	Easy/Good/Hard 버튼	학습자별 최적화 파라미터
난이도 모델	문제별 고정 (baseCorrectRate)	카드별 Ease factor	카드별 Difficulty
간격 계산	연속 함수 (지수 감쇄)	정수 배수 (×2.5 등)	연속 함수 (power law)
결과 입력	3단계 (정답/부분/오답)	4단계 (Again/Hard/Good/Easy)	4단계
복잡도	낮음 (파라미터 6개)	중간	높음 (파라미터 19개)

구현 위치

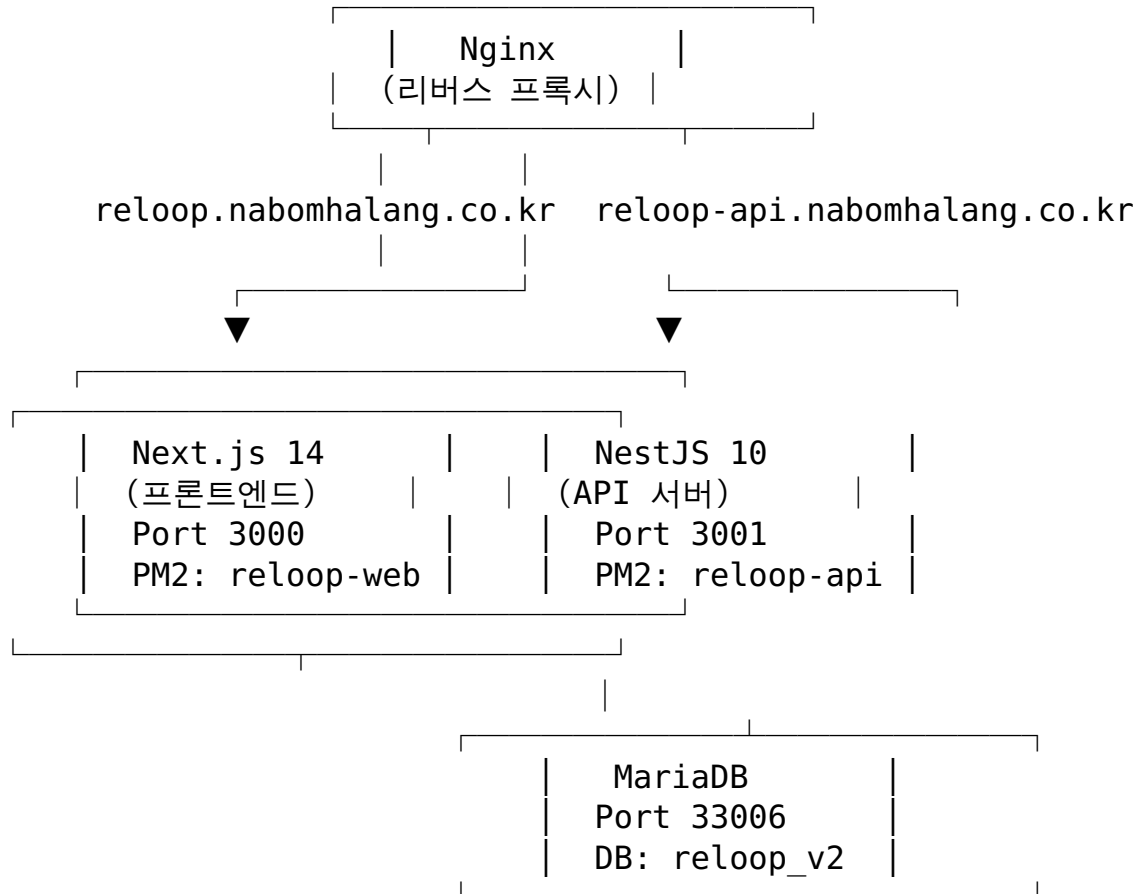
- backend/src/forget/persona-forget.service.ts — 알고리즘 핵심 클래스
- backend/src/reviews/reviews.service.ts — 복습 큐 관리 + 스케줄 생성
- backend/src/study-logs/study-logs.service.ts — 학습 기록 생성 시 S₀ 업데이트 + ReviewSchedule 생성

향후 개선 방향

1. **per-user λ** 학습: 사용자의 실제 복습 결과를 축적해 λ를 자동 조정 (현재는 페르소나로 고정)
2. **FSRS** 하이브리드: power-law 기반 감쇄를 옵션으로 추가
3. 태그별 λ: 같은 사용자라도 과목/태그에 따라 다른 망각 속도 적용
4. 최적 복습량 상한: 하루 복습 큐가 너무 길어지지 않도록 우선순위 기반 필터링

ReLoop 시스템 아키텍처

전체 구성



백엔드 모듈 구조

인증 (auth/)

- JWT 기반 stateless 인증
- POST /auth/register — 회원가입 (bcrypt 해시)
- POST /auth/login — 로그인 (JWT 발급, 7일 만료)
- GET /auth/me — 현재 사용자 정보 (JwtAuthGuard)
- 모든 보호 라우트는 @UseGuards(JwtAuthGuard)로 보호

프로필 (me/)

- PATCH /me/profile — 닉네임, 페르소나, 등급, 리뷰 민감도, 목표 연도, 중점 과목 등 수정
- POST /me/avatar — 아바타 이미지 업로드 (multipart/form-data, 최대 2MB, jpeg/png/webp)

- 업로드 파일은 backend/uploads/avatars/에 저장, /uploads/avatars/ URL로 서빙

문제집 (**problem-sets/**)

- GET /problem-sets — 문제집 목록 (년도, 과목 필터 가능)
- GET /problem-sets/:id — 문제집 상세 (문항 + 지문 포함)
- 문제집은 공유 자원 (모든 사용자가 동일한 문제집 접근)

학습 기록 (**study-logs/**)

- POST /study-logs — 개별 학습 기록 생성
- POST /study-logs/from-problem-set — 모의고사 일괄 제출 (자동 채점 + Subject/Tag 자동 생성 + ReviewSchedule 생성)
- GET /study-logs — 내 학습 기록 목록
- GET /study-logs/:id — 상세 (문제 정보 + 리뷰 스케줄 포함)
- PATCH /study-logs/:id — 수정 (chosenAnswer, result, memo)

복습 (**reviews/**)

- GET /reviews/queue — 오늘의 복습 큐 (scheduledAt이 현재 이전인 ReviewSchedule 조회)
- POST /reviews/:id/submit — 복습 결과 제출 → So 업데이트 → 다음 ReviewSchedule 생성
- 복습 큐는 PersonaForgetService.schedule()이 결정한 날짜에 따라 자동으로 채워짐

망각 곡선 엔진 (**forget/**)

- PersonaForgetService — 순수 계산 서비스 (DB 접근 없음)
- updateSo(input) — 결과에 따른 실력 업데이트
- schedule(input) — 다음 복습 시점 계산
- predictP(opts) — 특정 시점의 정답 확률 예측 (통계용)
- sampleCurve(opts) — 시간에 따른 실력/확률 곡선 샘플링 (차트용)

통계 (**stats/**)

- GET /stats/subjects — 과목별 통계 (So 평균, 태그 수)
- GET /stats/mastery-path — 숙련도 성장 곡선 데이터 (태그별 시계열)

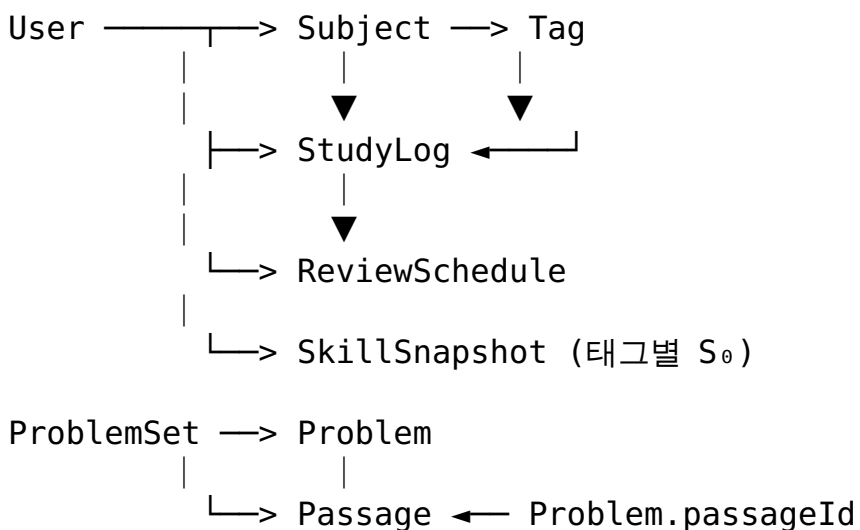
보안

- helmet — HTTP 보안 헤더
- @nestjs/throttler — 전역 60req/min, 인증 엔드포인트 10req/min
- CORS whitelist (CORS_ORIGINS 환경변수)
- ValidationPipe — 요청 body 화이트리스트 + 자동 변환

프론트엔드 라우트 맵

경로	인증	설명
/	공개	랜딩 페이지 (로그인 시 대시보드 리다이렉트)
/login	공개	로그인
/register	공개	회원가입
/pricing	공개/로그인	플랜 (비로그인: 마케팅, 로그인: 현재 플랜 표시)
/privacy /terms /contact	공개	법적 페이지
/onboarding	로그인	신규 사용자 온보딩 (등급, 과목, 페르소나)
/dashboard	로그인	대시보드 (채움/빈 상태 자동 분기)
/exams	로그인	문제집 카탈로그
/study/exam/[id]	로그인	모의고사 시험 화면
/study/exam/[id]/result	로그인	시험 결과
/study	로그인	학습 기록 직접 입력
/study/history	로그인	학습 기록 히스토리
/study-logs/[id]	로그인	학습 기록 상세
/solve/[id]	로그인	개별 문제 풀기
/review	로그인	복습 세션
/stats	로그인	통계 (히트맵, 정답률, 성장 곡선)
/subjects	로그인	과목 및 태그 관리
/profile	로그인	프로필 + 학습 설정 + 알림 + 계정

DB 스키마 (ER)



주요 모델

- **User**: 이메일, 비밀번호(bcrypt), 닉네임, 페르소나, 등급, 복습 민감도, 구독 티어
- **Subject**: 사용자별 과목 (이름, 색상). 과목 안에 Tag 목록
- **Tag**: 과목 하위 태그 (예: 수학 > 미적분). SkillSnapshot과 1:1
- **StudyLog**: 학습 기록 한 건 (결과, 난이도, 메모, 시간, 선택한 답)
- **ReviewSchedule**: 복습 스케줄 (예정일, 상태, 반복 회차, 예측 정답률)
- **SkillSnapshot**: 태그별 최신 So (실력 수치)
- **ProblemSet**: 문제집 (제목, 시험 유형, 연도, 과목명, 오디오 경로)
- **Problem**: 개별 문항 (번호, 본문, 선택지, 정답, 이미지 URL, 난이도)
- **Passage**: 공통 지문 (본문, 범위, 이미지 URL)

ReLoop 사용자 가이드

시작하기

1. 회원가입

1. <https://reloop.nabomhalang.co.kr> 접속
2. “무료로 시작하기” 또는 상단 “로그인” 클릭
3. 회원가입 탭에서 이메일, 닉네임, 비밀번호 입력
4. 체험만 하고 싶다면 “로그인 없이 체험하기”로 데모 계정 사용 가능

2. 온보딩

회원가입 직후 4단계 온보딩이 진행됩니다:

1. 목표 설정 — 닉네임 확인 + 목표 시험 선택 (수능/자격증/어학/없음)
2. 현재 실력 — 현재 등급과 목표 등급 설정 (1-9등급 카드 선택)
3. 중점 과목 — 학습에 집중할 과목 2-4개 선택
4. 학습 리듬 — 페르소나 선택 (뒤에 자세히 설명)

핵심 기능

대시보드

로그인 후 가장 먼저 보는 화면입니다.

- 오늘의 복습 큐: 오늘 복습해야 할 문항 수. “지금 시작” 버튼으로 바로 진입
- 주간 정답률: 이번 주에 복습한 문항의 정답률 추이
- 숙련도 진행: 전체 태그 중 90% 이상 숙련된 비율 (원형 게이지)
- 곧 마스터할 영역: 숙련도 80% 이상인 태그들 (조금만 더 하면 마스터)
- 최근 학습: 최근에 풀었거나 복습한 문항 목록 (클릭하면 상세 보기)
- 학습 코치: 현재 페르소나에 맞는 조언 메시지

문제집 (/exams)

사이드바 “문제집” 메뉴에서 기출 문제집 카탈로그를 볼 수 있습니다.

- 연도별, 과목별 필터로 원하는 시험지 검색
- “시작” 버튼을 누르면 모의고사 모드로 진입
- 현재 제공 중인 문제집: 2025/2026 수능 국어, 수학, 영어, 한국사, 생활과 윤리

모의고사 모드

실제 시험처럼 문제를 풀 수 있습니다.

화면 구성: - 상단: 시험 제목 + 카운트다운 타이머 + 진행률 - 중앙: 문제 이미지 (원본 시험지에서 자동 크롭) + 공통 지문 (접기/펼치기) - 하단: OMR 답안 버블 (①②③④⑤) + 네비게이션 버튼

기능: - 이전 / 건너뛰기 / 다음 / 항목이동 (4개 버튼) - 문제 이미지 클릭 → 확대 보기 - 키보드 단축키: 1-5(답 선택), ←/→(문제 이동), F(표시), Enter(다음) - 영어 듣기: 오디오 플레이어 상단에 표시 (자동 트랙 매칭) - 시험 종료: 우상단 : 메뉴에서 “시험 종료(제출)” 선택

집중 모드: - 기본: 사이드바가 보이는 일반 모드 (OFF) - 우상단 “집중 모드” 버튼으로 전체 화면 모드 전환 가능

제출 후: - 자동 채점 결과: 맞은 문제 수 / 총 문항 + 정답률 - 각 문항의 내 답 vs 정답 비교 - 모든 문항이 학습 기록에 자동 저장 → 복습 큐에 자동 진입

복습 세션 (/review)

대시보드의 “지금 시작” 또는 사이드바 “복습” 메뉴로 진입합니다.

- 오늘 복습해야 할 문항이 카드 형태로 순서대로 제시
- 문제 이미지를 보고 스스로 답을 떠올린 뒤 결과를 기록:
 - 맞음: 실력(So)이 크게 상승, 다음 복습까지 간격 늘어남
 - 부분: 약간 상승, 간격 약간 늘어남
 - 틀림: 실력 하락, 짧은 간격으로 재복습
 - 건너뛰기: 이 카드를 나중에 미룸
- 결과 기록 후 2초간 “결과 기록됨” 토스트 표시 (실행 취소 가능)
- 모든 카드 처리 시 완료 축하 화면

학습 기록 (/study/history)

모든 학습 기록을 테이블(데스크톱) 또는 카드 리스트(모바일)로 확인합니다.

- 과목, 날짜, 결과로 필터링
- 검색창에서 제목/과목/태그로 검색
- 각 항목 클릭 → 상세 페이지에서:
 - 문제 이미지 + 내 답 vs 정답
 - 복습 스케줄 타임라인

- “다시 풀기” 버튼 (개별 문제 풀기 화면으로 이동)

통계 (/stats)

학습 패턴과 성장을 시각적으로 확인합니다.

- 학습 히트맵: 최근 18주간 날짜별 학습량 (색상 농도)
- 과목별 정답률: 각 과목의 정답/오답 비율 바 차트
- 숙련도 성장 곡선: 태그별 숙련도가 시간에 따라 어떻게 변했는지 선 그래프

과목 및 태그 (/subjects)

과목과 태그를 관리합니다.

- 과목별로 하위 태그 목록 + 각 태그의 현재 숙련도
- 태그 이름 수정 (인라인 편집)
- 태그 보관 (소프트 삭제 + 실행 취소)
- 새 과목/태그 추가

프로필 (/profile)

4개 탭으로 구성:

탭	내용
프로필	아바타 업로드, 닉네임, 이메일, 현재 플랜
학습 설정	현재/목표 등급 (카드 선택), 페르소나, 복습 민감도
알림	일일 복습 알림, 주간 리포트, 마케팅 수신 토글
계정	플랜 관리, 데이터 내보내기, 진행 초기화, 로그아웃

페르소나 선택 가이드

페르소나	추천 대상	복습 빈도
숙련자 (Senior)	이미 기초가 탄탄한 사람. 유지 목적 학습	주 1-2회
중급 (Mid)	대부분의 학습자. 추천 기본값	2-3일마다
초보 (Junior)	새로운 분야를 시작한 사람	매일~격일
단기집중 (Crammer)	시험 직전 벼락치기	매일 여러 번

나중에 프로필 설정에서 언제든지 변경할 수 있습니다.

모바일 사용

- 하단 네비게이션 바로 주요 메뉴 접근
- 프로필 탭에서 설정 + 로그아웃
- 모의고사: 문제 이미지 탭하면 확대
- PWA 설치: 브라우저 메뉴에서 “홈 화면에 추가”로 앱처럼 사용 가능

자주 묻는 질문

Q: 복습 큐에 너무 많은 문제가 쌓여요. **A:** 프로필 → 학습 설정에서 복습 민감도를 “여유”로 변경하면 복습 간격이 길어집니다. 또는 페르소나를 “숙련자”로 바꿔보세요.

Q: 수능 외에 다른 공부에도 쓸 수 있나요? **A:** 네. 과목과 태그를 자유롭게 만들어 자격증, 어학, 취미 학습 등 어떤 공부든 기록하고 복습할 수 있습니다. KICE 기출 문제집은 수능 준비생을 위한 추가 기능입니다.

Q: 오답 노트처럼 쓸 수 있나요? **A:** 학습 기록 상세에서 메모를 남길 수 있고, 틀린 문항은 자동으로 짧은 간격의 복습 큐에 들어갑니다. “다시 풀기”로 해당 문제만 다시 도전할 수도 있습니다.

Q: 데이터를 백업할 수 있나요? **A:** 프로필 → 계정 탭에서 “데이터 내보내기”를 누르면 학습 기록을 JSON 파일로 다운로드할 수 있습니다.

ReLoop 운영 및 배포 가이드

서버 환경

항목	값
OS	Debian 12 (VM on Proxmox)
Node.js	20+
pnpm	9.12+
DB	MariaDB (Docker, Port 33006)
프로세스 매니저	PM2
리버스 프록시	Nginx
도메인	reloop.nabomhalang.co.kr (프론트), reloop-api.nabomhalang.co.kr (API)

배포 절차

1. 코드 동기화

```
# Gitea에서 pull
cd /home/dev/reloop-v2
git fetch origin && git reset --hard origin/master
```

2. 백엔드 배포

```
cd backend
pnpm install --no-frozen-lockfile
```

```
pnpm prisma:deploy # 마이그레이션 적용 (안전, 데이터 보존)
rm -f tsconfig.build.tsbuildinfo # 빌드 캐시 초기화
pnpm exec tsc -p tsconfig.build.json # 컴파일
pm2 restart reloop-api
```

pnpm build (nest build) 대신 tsc -p tsconfig.build.json을 직접 사용합니다. nest CLI의 빌드 래퍼가 간헐적으로 dist/ 를 비우는 버그가 있어서 직접 tsc를 실행하는 것이 안정적입니다.

3. 프론트엔드 배포

```
cd frontend
pnpm install --no-frozen-lockfile
pnpm build # Next.js 프로덕션 빌드
cp -r .next/static .next/standalone/.next/
cp -r public .next/standalone/ 2>/dev/null || true
pm2 restart reloop-web
```

4. 상태 확인

```
pm2 list | grep reloop
# reloop-api online
# reloop-web online
```

헬스체크

```
curl -s -o /dev/null -w "%{http_code}\n" https://reloop-api.nabomhalang.co.kr/api/health
curl -s -o /dev/null -w "%{http_code}\n" https://reloop.nabomhalang.co.kr/
```

KICE 기출 데이터 관리

PDF 원본 배치

```
data/kice/
├── 2025/
│   ├── 국어/
│   │   ├── (25)국어영역_문제지_홀수형.pdf
│   │   └── (25)국어영역_정답표.pdf
│   ├── 수학/
│   ├── 영어/
│   │   ├── (25)영어영역_문제지_홀수형.pdf
│   │   ├── (25)영어영역_정답표.pdf
│   │   └── 영어영역듣기평가음원/*.mp3
│   ├── 한국사/
│   └── 사회탐구/사회탐구영역_문제지/01 생활과 윤리_문제지.pdf
```

└── 2026/
 └── (같은 구조)

임포트 실행

```
cd backend
```

```
# 특정 과목
```

```
pnpm cli:kice-import --year=2026 --subject=국어
```

```
# 특정 연도 전체
```

```
pnpm cli:kice-import --year=2026
```

```
# 전체 (2025 + 2026)
```

```
pnpm cli:kice-import
```

```
# 드라이 런 (DB 변경 없이 파일 스캔만)
```

```
pnpm cli:kice-import --dry-run
```

임포트가 하는 일: 1. PDF 페이지를 문제별로 PNG 크롭 → uploads/problems/{year}/{subject}/{number}.png 2. 정답표 파싱 (텍스트 또는 OCR) 3. ProblemSet / Problem / Passage DB upsert (먹등)

OCR 보정 (2025 정답표 등)

2025 정답표는 이미지 기반 PDF라 텍스트 추출이 안 됩니다. Codex Vision OCR로 보정:

```
pnpm cli:ocr-fix --year=2025 --answers-only    # 정답만 채움  
pnpm cli:ocr-fix --year=2025                  # 문항 본문까지 보정  
pnpm cli:ocr-fix --dry-run                     # 미리보기
```

codex CLI가 설치되어 있어야 하고 ChatGPT 로그인이 되어 있어야 합니다.

문제 이미지 서버 전송

임포트는 로컬에서 실행하고 이미지는 서버로 전송:

```
# 로컬에서
```

```
cd backend
```

```
tar -czf /tmp/problem-images.tar.gz -C uploads problems/
```

```
# 서버에서
```

```
cd /home/dev/reloop-v2/backend/uploads
```

```
tar -xzf /tmp/problem-images.tar.gz
```

PM2 ecosystem 설정

ecosystem.config.js 예시:

```
module.exports = {
  apps: [
    {
      name: 'reloop-api',
      script: 'dist/main.js',
      cwd: '/home/dev/reloop-v2/backend',
      env: { NODE_ENV: 'production', PORT: 3001 },
    },
    {
      name: 'reloop-web',
      script: '.next/standalone/server.js',
      cwd: '/home/dev/reloop-v2/frontend',
      env: { NODE_ENV: 'production', PORT: 3000, HOSTNAME: '0.0.0.0' },
    },
  ],
};
```

모니터링

```
pm2 logs reloop-api --lines 50      # API 로그
pm2 logs reloop-web --lines 50      # 프론트 로그
pm2 monit                            # 실시간 CPU/메모리
```

DB 관리

```
# Prisma Studio (로컬에서)
cd backend && pnpm exec prisma studio
```

```
# 직접 SQL
mysql -h 10.10.10.146 -P 33006 -u reloop_v2 -p reloop_v2
```

```
# 마이그레이션 상태 확인
pnpm exec prisma migrate status
```

```
# 데이터 초기화 (주의!)
pnpm prisma:reset      # 전체 드롭 + 재생성 + 시드
```

보안 체크리스트

- ☐ JWT_SECRET이 32자 이상 랜덤 문자열인지 확인
- ☐

CORS_ORIGINS에 프론트엔드 도메인만 허용되어 있는지 확인



.env 파일이 git에 커밋되지 않았는지 확인



rate limit 동작 확인:

```
for i in {1..70}; do curl -s -o /dev/null -w "%{http_code} "
https://reloop-api.nabomhalang.co.kr/api/auth/login; done
```



HTTPS가 nginx에서 강제되는지 확인



uploads/ 디렉토리 권한이 644 (파일) / 755 (디렉토리)인지 확인

트러블슈팅

증상	원인	해결
reloop-api 즉시 크래시	dist/main.js 없음	tsc -p tsconfig.build.json 실행
express 모듈 못 찾음	pnpm strict 모드	pnpm install 재실행 (express가 deps에 명시됨)
프론트 빌드 실패 (@phosphor-icons)	패키지 미설치	pnpm install --no-frozen-lockfile
문제 이미지 404	uploads 디렉토리 없음	mkdir -p uploads/ problems + 이미지 전송
2025 정답 null	이미지 PDF + OCR 미실행	pnpm cli:ocr-fix -- year=2025 --answers-only
마이그레이션 실패 (shadow DB)	INDEX 권한 없음	MariaDB root로 GRANT INDEX ON *.* TO 'reloop_v2'@'%'

ReLoop PDF 파싱 파이프라인

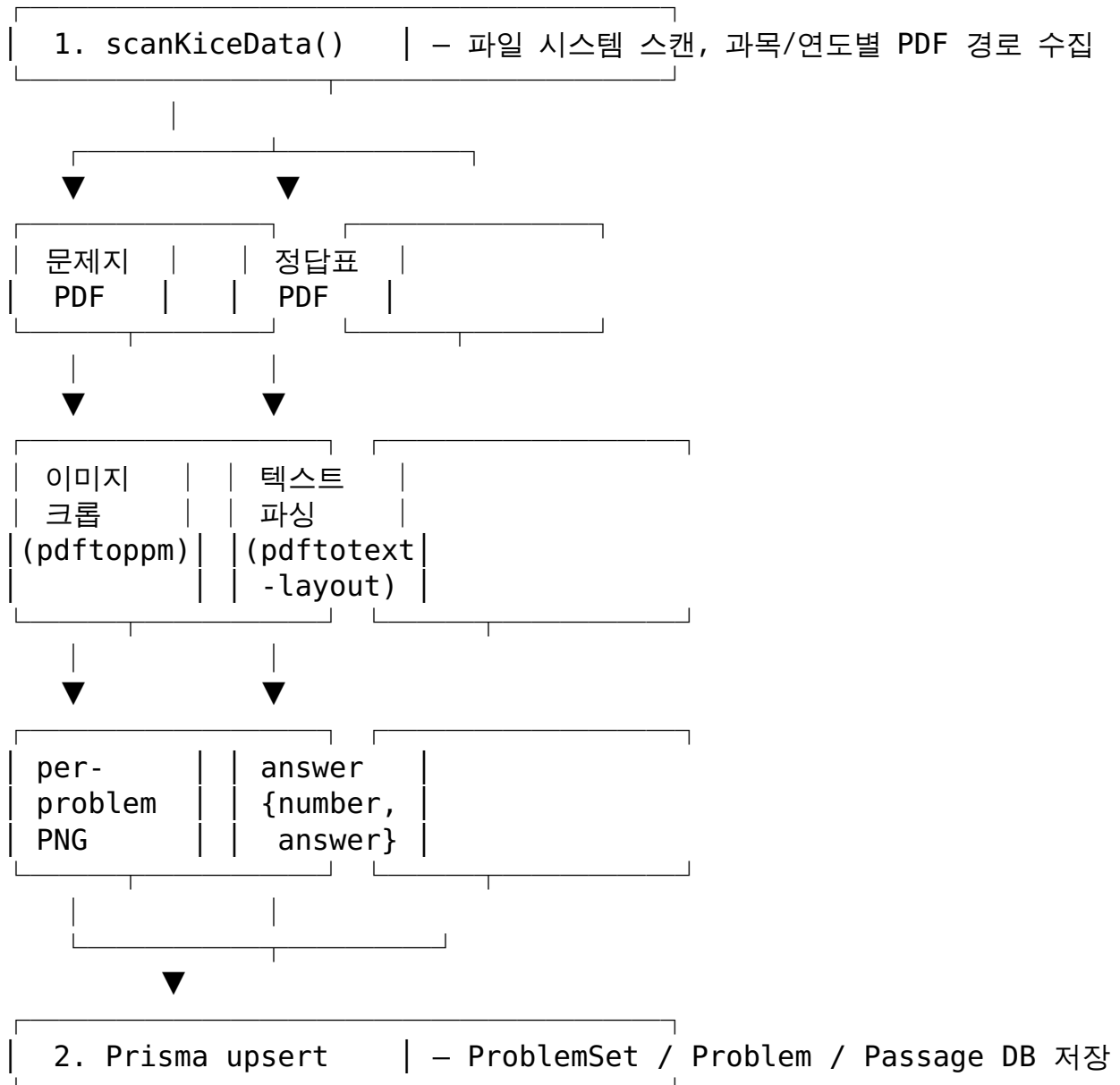
개요

ReLoop은 KICE(한국교육과정평가원) 수능 기출 PDF를 자동으로 파싱해 문제집 데이터로 변환합니다. 텍스트 추출과 이미지 크롭을 병행하며, 이미지가 주(primary) 표시이고 텍스트는 보조(검색/채점용)입니다.

파이프라인 흐름

KICE PDF 파일





도구

도구	용도	패키지
pdftinfo	페이지 수, 페이지 크기 확인	poppler-utils
pdftotext -bbox	단어별 바운딩 박스 XML 추출 (문제 번호 Y좌표 감지)	poppler-utils
pdftotext -layout	정답표 텍스트 추출 (표 형태 보존)	poppler-utils
pdftotext -raw	문제지 텍스트 추출 (보조, 검색용)	poppler-utils
pdftoppm	PDF 페이지를 PNG 이미지로 렌더링 + 크롭	poppler-utils
codex exec -i	이미지 기반 OCR (2025 정답표 등)	codex CLI

핵심 모듈

```
backend/src/problem-sets/parsing/
├── index.ts                # parseExamPaper() - 공개 API
├── types.ts                # ParseResult, ParsedProblem 등 타입
입
├── extract-text.ts         # pdftotext 래퍼
├── strip-page-chrome.ts    # 페이지 헤더/푸터 제거
├── parse-answer-table.ts   # 정답표 파서
├── parse-problem-paper.ts  # 문제지 텍스트 파서 (보조)
├── parse-image-based-exam.ts # 문제별 이미지 크롭 (주)
├── strategies/
│   ├── kice.ts             # KICE 수능 전용 규칙
│   └── math.ts             # 수학 과목 전용 규칙
├── ocr-fallback/
│   └── index.ts            # ocrAnswerTable(),
ocrProblemPage()
│   ├── render-page.ts     # pdftoppm 래퍼
│   ├── codex-vision.ts    # codex exec -i 래퍼
│   └── prompts/           # OCR 프롬프트 템플릿
└── README.md
```

이미지 크롭 알고리즘

문제 번호 위치 감지

1. 각 페이지에 `pdftotext -bbox` 실행 → XML 출력
2. XML에서 `\d{1,2}\.` 패턴의 단어 위치 추출 (`yMin`, `xMin`)
3. `[M~N]` 형태의 지문 범위 마커도 같이 추출

크롭 영역 계산

페이지 크기: 842pt × 1191pt (A3)

렌더 DPI: 300

변환 비율: $\text{pixel} = \text{point} \times 300/72 \approx \text{point} \times 4.167$

문제 N의 크롭 영역:

Y 시작 = (문제 N의 `yMin` - 15pt 여유) × 4.167

Y 끝 = (문제 N+1의 `yMin` - 5pt) × 4.167 (마지막 문제면 페이지 하단)

X 시작 = 0

너비 = 페이지 전체 너비 × 4.167

최소 높이 = 200px (너무 작은 크롭 방지)

pdftoppm 크롭 실행

```
pdftoppm -r 300 -png -f {page} -l {page} \  
-x 0 -y {startPx} -W {widthPx} -H {heightPx} \  
input.pdf /tmp/crop-prefix
```

출력: /tmp/crop-prefix-{page}.png

폴백

bbox에서 감지되지 않은 문제 번호는 해당 페이지 전체를 이미지로 저장합니다.

정답표 파싱

텍스트 기반 (2026)

```
pdftotext -layout input.pdf -
```

출력 예:

```
1   ③   2   18   ②   2  
2   ⑤   2   19   ④   2
```

정규식 `/(\d{1,2})\s*([\①②③④⑤])/g` 으로 추출, 원문자를 1-5로 변환.

OCR 기반 (2025)

2025 정답표는 PScript5.dll + Acrobat Distiller로 생성되어 텍스트 레이어가 없습니다.

1. pdftoppm -r 300 으로 페이지 렌더링
2. codex exec --sandbox read-only -i page.png 으로 Vision API 호출
3. 구조화된 JSON 응답 파싱: [{number, answerNumber}, ...]
4. Problem 레코드에 answerNumber 업데이트

페이지 헤더/푸터 제거 (**strip-page-chrome.ts**)

pdftotext 결과에는 페이지 헤더/푸터가 포함됩니다. 이를 제거하지 않으면 다음 문제의 본문에 섞입니다.

제거 대상 라인 패턴: - `^\s*\d+\s*$` — 페이지 번호만 있는 줄 - `^\s*\d+\s+홀수형\s*$` — “2 홀수형” - `^\s*제\s*\d+\s*교시` — “제1 교시” - `^\s*\d{4}학년도.*대학수학능력시험.*문제지` — 시험 제목 - `^\s*이 문제지에 관한 저작권은` — 저작권 문구 - `\f` — 폼 피드 문자

전략 패턴 (확장성)

새로운 시험 형식 (EBS, 사설 모의고사 등)을 추가하려면:

1. strategies/ 에 새 파일 생성 (예: ebs.ts)
2. PageChromeStrategy 인터페이스 구현 (헤더/푸터 정규식 패턴)
3. parseExamPaper() 호출 시 format: 'ebs' 전달

현재 구현된 전략: - kice — KICE 수능 (기본) - math — 수학 과목 (이미지 전용) - generic — 알 수 없는 형식 (최소 strip + 기본 정규식)

출력 데이터

Problem 레코드

필드	값	설명
imageUrl	/uploads/problems/2026/국어/001.png	문제 크롭 이미지 (주 표시)
pageImageUrl	/uploads/problems/2026/국어/page-1.png	전체 페이지 (폴백)
bodyText	윗글의 내용과 일치하지 않는 것은?	텍스트 추출 (검색/접근성)
choices	{"1": "...", "2": "...", "3": "...", "4": "...", "5": "..."}	선택지 (보조)
answerNumber	3	정답 (자동 채점용)
needsReview	false	파싱 품질 플래그

Passage 레코드

필드	값
imageUrl	/uploads/problems/2026/국어/passage-1-3.png
bodyText	[1~3] 다음 글을 읽고 물음에 답하시오. ...
startNumber	1
endNumber	3

CLI 명령어 요약

```
# 전체 임포트 (스캔 + 크롭 + 파싱 + DB upsert)
pnpm cli:kice-import
```

특정 대상

```
pnpm cli:kice-import --year=2026 --subject=국어
```

드라이 런

```
pnpm cli:kice-import --dry-run
```

OCR 보정

```
pnpm cli:ocr-fix --year=2025 --answers-only
```

```
pnpm cli:ocr-fix --year=2025 --subject=수학
```

검증

```
pnpm exec ts-node -e "
```

```
import { PrismaClient } from '@prisma/client';
```

```
const p = new PrismaClient();
```

```
(async () => {
```

```
  const total = await p.problem.count();
```

```
  const withImg = await p.problem.count({ where: { imageUrl:
    { not: null } } });
```

```
  console.log(withImg + '/' + total + ' problems with images');
```

```
  await p.$disconnect();
```

```
})();
```

```
"
```