

퀄리저널 관리자 시스템 개선 작업 인수인계 보고서

이 문서는 퀄리저널(QualiJournal) 관리자 시스템의 기능 개선 작업 현황과 향후 계획을 정리한 업무 인수인계 보고서입니다. 이전 채팅에서 논의되었던 개선 사항들을 바탕으로, 새로운 채팅 세션에서도 일관된 맥락을 유지하며 작업을 이어나갈 수 있도록 현재까지의 진행 상황과 남은 과제들을 체계적으로 정리합니다. 주요 요구 사항 및 구현해야 할 기능들을 목록으로 정리하였으며, 특히 **보고서 생성/요약/내보내기 기능**(항목 3)을 다음 작업으로 선정하여 상세 계획을 설명합니다. 또한 통합 테스트와 보안 강화 등의 추가 작업 계획과 장기 개선 방향도 함께 포함합니다.

주요 작업 항목 및 요구 사항

프로젝트 **중기 실행계획서**와 관련 설계 문서를 통해 도출된 주요 개선 과제는 다음과 같습니다. 각 항목은 구현하거나 개선이 필요한 기능과 세부 요구 사항을 나타냅니다:

- 게이트 슬라이더** - UI에 게이트 임계값을 조정할 수 있는 슬라이더를 추가하고, 값 변경 시 `/api/config/gate_required`에 PATCH 요청을 보내도록 합니다. (패치 설계서에서는 슬라이더 제공과 KPI 자동 갱신을 요구하고 있습니다.)
- KPI 자동 새로고침** - 현재 KPI(선정본 수, 승인 수 등)는 ‘새로고침’ 버튼을 눌러야 갱신됩니다. **30초 간격**으로 `/api/status`를 호출하여 자동으로 KPI를 갱신하는 타이머를 추가해야 합니다.
- 보고서·요약·내보내기 API 및 UI** - 중기 실행계획서에서 `/api/report` (POST), `/api/enrich/keyword` / `/api/enrich/selection` (POST), `/api/export/md` / `/api/export/csv` (GET) 엔드포인트를 구현하고 UI에 대응 버튼들을 추가할 것이 요구되었습니다. 현재 서버에는 `GET /api/report`과 `GET /api/export/{fmt}`만 일부 구현되어 있으며, **보고서 생성**이나 **카드 요약·번역** 기능과 이를 트리거하는 UI 버튼은 아직 존재하지 않습니다.
- 보고서 생성 엔드포인트 추가** - 상기 기능을 지원하기 위해 백엔드에 `POST /api/report` 엔드포인트를 추가하고, `tools/make_daily_report.py` 스크립트를 호출하여 `archive/reports/{date}_{keyword}.md` 형식의 일간 보고서(markdown)를 생성하도록 합니다. 생성이 완료되면 해당 파일 경로를 JSON 형태로 반환하고, 프론트엔드 UI에 **보고서 링크**를 표시합니다.
- 카드 요약·번역 엔드포인트 추가** - 서버에 `POST /api/enrich/keyword`와 `POST /api/enrich/selection` 엔드포인트를 추가하여 `tools/enrich_cards.py` 스크립트를 호출하고, 기사 **요약 및 번역** 결과 파일 경로를 반환합니다. 프론트엔드 UI에도 “**카드 요약/번역**” 버튼을 추가하여, 클릭 시 해당 API를 호출하고 반환된 결과 파일 링크를 보여줍니다.
- 내보내기 UI 개선** - 서버의 `GET /api/export/md`와 `GET /api/export/csv` 엔드포인트를 호출하는 **내보내기** 버튼들을 UI에 배치하고, 현재 발행본 데이터를 **Markdown 파일**이나 **CSV 파일**로 다운로드할 수 있도록 합니다. (현재는 API 엔드포인트만 있을 뿐 UI에서 이를 사용하는 기능이 없는 상태입니다.)
- 테스트 및 문서화** - Mid-term plan 3주차와 4주차 계획에는 **Pytest 기반 통합 테스트** 작성과 **README 문서** 갱신이 포함되어 있습니다. 새로 추가된 API들과 UI 동작을 검증하는 테스트 스크립트를 작성하고, **비동기 실행** (태스크 처리) 방식, **폴백 동작**(구 버전 API 대응), **게이트 임계값 조정 방법** 등을 README에 추가하여 문서화해야 합니다.

8. **보안 강화 (추가 사항)** - 패치 설계서에서는 **토큰 기반 인증, HTTPS 적용, CORS 정책 강화** 등을 권장합니다. 가능하다면 **토큰 검증 미들웨어**를 서버에 추가하여 모든 API 호출 시 유효한 토큰이 있는지 확인하고, (추후 구현을 대비하여) **관리자와 일반 사용자** 권한을 구분하는 준비를 시작합니다. 이 항목은 외부 인증 시스템과의 연동이 필요한 장기 과제이므로 우선순위는 낮지만, **당일 중으로 토큰 검증 미들웨어의 초안을 적용**해 보는 것이 목표입니다.
9. **엔진/오케스트레이터 고도화 및 AI 요약 (장기 과제)** - 기능목표 설계 문서에서는 **키워드 역사 큐레이션, 트렌드 분석 자동화, AI 요약/번역 고도화** 등 향후 적용할 수 있는 장기 개선 사항을 제안하고 있습니다. 이러한 기능들은 이번 중기 개선 범위에는 포함되지 않으므로 즉각 구현 대상은 아니나, 향후 로드맵에 **추가 개선 과제**로서 기록합니다.

현재 진행 현황

현재까지 **1번과 2번 작업**에 대한 대비가 이루어졌습니다. UI 상단에 **게이트 임계값**을 조절할 수 있는 슬라이더 컴포넌트를 추가하였고, 슬라이더 값 변경 시 백엔드의 `/api/config/gate_required` PATCH 엔드포인트를 호출하여 `config.json`의 `gate_required` 값을 업데이트하는 기능을 구현했습니다. 이를 통해 편집자는 품질 게이트 임계값(예: 최소 승인 기사 수 15개)을 **실시간으로 조정**할 수 있게 되었습니다. 또한 KPI 지표(예: 수집된 기사 수, 승인된 기사 수 등)를 30초마다 자동 갱신하도록 프론트엔드에 **타이머 기반 갱신 로직**을 추가했습니다. 이제 별도의 수동 새로고침 버튼을 누르지 않아도 일정 주기마다 `/api/status`를 호출하여 최신 KPI 값을 대시보드에 반영합니다. 이 두 가지 개선(동적 게이트 임계값 설정, KPI 자동 새로고침)은 **패치 설계 요구사항**을 충족하며, 편집 작업의 편의성과 즉시성 향상에 기여했습니다.

한편, **보고서 생성, 요약·번역, 결과 내보내기**와 관련된 기능(항목 3~6)은 아직 구현되지 않은 상태입니다. 서버 측에 `/api/report`와 `/api/export` 일부 기능이 존재하지만 **POST 방식 보고서 생성**이나 **카드 요약/번역** 엔드포인트는 없는 상황입니다. 또한 현재 관리자 UI에는 보고서 생성이나 내보내기 등의 **버튼/메뉴가 존재하지 않아**, 관련 기능을 사용하지 못하고 있습니다. 이러한 부분은 중기 실행계획서에서 요구된 핵심 개선 사항으로, 다음 단계에서 우선적으로 개발이 필요합니다.

그 외에 **비동기 처리 엔드포인트**(`/api/tasks/...`)나 **실시간 스트리밍 진행 표시** 기능, **테마 통합 및 접근성 개선** 등의 작업이 병행되었거나 진행 중입니다. 단기 과제로 제시된 UI 다크/라이트 테마 통합과 폴백(fallback) 기능, 인코딩 오류 수정 등의 항목은 이미 이전 단계에서 완료되었거나 적용되고 있습니다. 이제 남은 주요 기능 구현과 중기 과제들의 마무리를 위해, 아래와 같이 **보고서·요약·내보내기 기능 추가**를 비롯한 후속 작업을 계획대로 추진해야 합니다.

다음 단계: 보고서·요약·내보내기 기능 구현 (항목 3 중심)

이제 **가장 시급한 과제로 보고서 생성, 카드 요약·번역, 결과물 내보내기** 기능을 구현할 계획입니다. 이는 위의 항목 **3, 4, 5, 6번**에 해당하는 내용으로, 중기 계획의 핵심 요구사항 중 하나입니다. 현재 관리자 시스템에는 이 기능들이 비어 있으므로, 백엔드 API와 프론트엔드 UI 양측을 모두 확장하여 편집자가 **일간 보고서를 생성**하고, **뉴스 카드를 요약/번역**하며, **최종 선정된 기사 목록을 파일로 출력**할 수 있도록 할 것입니다.

- **백엔드 API 구현:** 우선 FastAPI 기반 서버에 새로운 엔드포인트들을 추가해야 합니다.
 - `POST /api/report` 엔드포인트를 구현하여 **일일 보고서 생성**을 트리거합니다. 이 엔드포인트는 내부적으로 `tools/make_daily_report.py` 스크립트를 호출하여 현재 선택된 기사들(`selected_articles.json`)을 바탕으로 Markdown 형식의 **데일리 보고서**를 생성합니다. 보고서 파일은 `archive/reports/{날짜}_{키워드}.md` 형태로 저장하고, 요청에 대한 응답으로 생성된 파일의 경로 또는 URL을 JSON으로 반환합니다. 마찬가지로, `POST /api/enrich/keyword`와 `POST /api/enrich/selection` 엔드포인트를 구현하여 **뉴스 카드 요약 및 번역**을 수행합니다. 이들은 `tools/enrich_cards.py` 스크립트를 호출하여 각각 **모든 수집 기사(keyword 기준)** 혹은 **편집자가 승인한 기사(selection 기준)**에 대한 요약/번역 작업을 수행하고, 그 결과를 파일로 저장한 뒤 경로를 반환합니다. 마지막

으로, `GET /api/export/md`와 `GET /api/export/csv` 엔드포인트를 완성시켜 현재 **발행본(final selection)** 데이터를 내려받을 수 있도록 합니다. Markdown의 경우 편집된 기사 목록을 Markdown 템플릿 형태로 제공하고, CSV의 경우 기사 목록을 CSV 형식으로 내보내는데, 이때 **UTF-8 BOM(Byte Order Mark)**을 포함시켜 Excel에서 열 때 인코딩 문제가 없도록 합니다. (`/api/export/csv` 응답에 BOM이 선행되도록 하여 Excel 호환성을 보장합니다.)

- **프론트엔드 UI 통합:** 새로운 백엔드 기능을 사용할 수 있도록 **관리자 UI에 버튼 및 연동 로직**을 추가합니다. 예를 들어, 보고서 생성을 위해 대시보드 화면에 **"일일 보고서 생성"** 버튼을 마련합니다. 사용자가 이 버튼을 클릭하면 `POST /api/report`를 호출하고, 응답으로 반환된 보고서 파일 경로를 받아 화면에 **다운로드 링크** 또는 **"보고서 열기"** 링크를 표시합니다. (필요하다면 보고서 내용을 모달이나 새 탭으로 표시할 수 있습니다.) 카드 요약/번역 기능 역시 **"카드 요약/번역"** 등의 버튼으로 제공하여, 클릭 시 해당 API(`/api/enrich/...`)를 호출하고 결과 파일 링크를 안내합니다. 요약/번역 결과물은 번역된 기사 요약본이 될 가능성이 높으므로, UI에서 열람할 수 있게 하거나 다운로드 받도록 합니다. 또한 **내보내기** 기능에 대해 UI에 **"Export MD", "Export CSV"** 버튼 혹은 드롭다운 메뉴를 추가합니다. 이 버튼들을 누르면 `/api/export/md` 또는 `/api/export/csv` 엔드포인트를 호출하여 바로 파일을 다운로드하도록 구현합니다. (예: `Export CSV` 형태로 링크를 구성하거나, JS로 API 응답을 받아 파일을 생성하는 방식 등을 고려합니다.) 이렇게 함으로써 편집자는 GUI 상에서 곧바로 최종 결과물을 마크다운이나 CSV 파일로 저장할 수 있게 됩니다.

- **상태 표시 및 예외 처리:** 보고서 생성이나 요약 작업은 다소 시간이 걸릴 수 있으므로, **진행 상황 표시나 사용자 피드백**도 고려해야 합니다. 간단한 구현으로는 버튼 클릭 시 로딩 인디케이터(spinner)를 보여주고, 응답이 도착하면 결과 링크를 표시하는 방식을 취합니다. 향후 비동기 작업 처리(`/api/tasks/...` 엔드포인트)가 도입되면 이러한 보고서/요약 생성도 **백그라운드 작업**으로 전환하고, **실시간 진행률**을 표시할 수 있겠지만, 현재 단계에서는 동기 처리로 구현하고 사용자에게 **완료 결과만** 알리는 형태로 충분합니다. 또한 예기치 않은 오류(예: 스크립트 실패 등)에 대비해 **오류 메시지**를 UI에 표시하거나 로그로 남겨 디버깅할 수 있도록 합니다.

이와 같이 **보고서·요약·내보내기 기능**을 구현함으로써, 중기 계획의 주요 기능적 공백을 메우고 편집 작업 흐름을 완성하게 됩니다. 특히 **일일 보고서** 생성은 퀴리저널 플랫폼의 산출물을 축적하고 공유하는 역할을 할 것이고, **뉴스 카드 자동 요약/번역**은 편집자의 수작업을 줄여 작업 효율을 높여줍니다. **Export 기능**은 자료를 백업하거나 다른 형식으로 활용할 수 있게 해주어 운영상의 편의를 증진시킵니다. 이러한 기능들이 추가되면, 관리자 시스템은 계획된 대로 **하루 한 키워드 뉴스**의 수집부터 발행, 보고서 아카이브까지 일련의 흐름을 UI 상에서 완결할 수 있게 됩니다.

통합 테스트 및 문서화 (항목 7)

모든 주요 기능 구현이 완료된 후에는 **통합 테스트 작성과 문서화 작업**을 수행해야 합니다. Mid-term 계획 3~4주차 일정에 명시된 대로, Pytest를 활용한 **테스트 스크립트**를 준비하여 새로운 API들과 기능들이 제대로 동작하는지 검증합니다. 예를 들어, `/api/report` 엔드포인트 호출 시 실제로 보고서 파일이 생성되고 예상 경로를 반환하는지 테스트하고, `/api/enrich/keyword` 호출 시 요약 결과가 나오는지, `/api/export/csv` 호출 결과로 받은 CSV 내용과 형식이 올바른지 등을 확인합니다. 프론트엔드에 대해서는 Selenium이나 Playwright 같은 도구까지는 아니더라도, 가능하면 **API 단위 테스트**와 함께 **간단한 시나리오 테스트**를 통해 (예: 가상으로 여러 기사를 수집하고 승인한 후 보고서 생성 -> 내보내기까지 흐름) 기능 통합을 검증합니다. 이러한 테스트들은 CI 파이프라인에 추가되어 **회귀 테스트** 역할을 할 수 있도록 준비합니다.

병행하여 **프로젝트 문서(README 등)**를 최신 상태로 업데이트해야 합니다. 새로운 기능들의 사용 방법과 구성 변경 내용을 반영하고, 운영 절차에 관한 설명을 보완합니다. 구체적으로, **비동기 실행 및 폴백 동작**에 대한 설명을 README에 추가하여, 현재 서버가 비동기 엔드포인트(`/api/tasks/...`)를 구현하지 못한 경우에도 기존 동기 엔드포인트(`/api/flow/...`)로 자동 폴백되는 구조를 관리자들이 이해할 수 있도록 합니다. 또한 **게이트 임계값 조정 방법** (예: config.json의 `gate_required` 설정과 UI 슬라이더 사용법), **KPI 자동 새로고침 기능** 등에 대해서도 README에 안내를 추가합니다. 이 외에도 환경 설정, 종속 패키지(requirements.txt) 목록, 폴더 구조(admin 폴더 등) 변화, 그리

고 새로 도입된 **API 엔드포인트 목록 및 설명**(예: `/api/report`, `/api/enrich/...`, `/api/export/...`)의 용도와 요청/응답 형식)을 문서화합니다. 마지막으로 **운영 매뉴얼** 차원의 내용(예: 승인 임계값 변경 시 고려사항, 키워드 입력 및 발행 절차, 주간 회귀테스트 방법 등)을 정리하여 인수인계 자료로 남깁니다.

문서화와 테스트 작업은 기능 구현만큼 중요하며, 특히 이번 개선으로 시스템 구조(비동기 처리, 설정값 노출 등)가 많이 바뀌는 만큼 차후 유지보수와 신규 인력 투입 시 원활한 이해를 위해 자세히 기록해야 합니다. 주어진 일정 내에 테스트와 문서 작업을 완료함으로써, 중기 목표 범위의 과제를 성공적으로 마무리할 수 있을 것입니다.

보안 강화 계획 (항목 8)

보안 관련 개선은 **장기적 과제**로 분류되어 있지만, 가능하다면 일부 작업을 선제적으로 착수합니다. 패치 설계 문서와 실행계획서에서는 **토큰 기반 인증** 도입과 **HTTPS 적용**, 그리고 **CORS 정책 강화**를 권장하고 있습니다. 현재 시스템에는 인증이나 권한 관리가 전무한 상태이므로, 우선 **토큰 인증 미들웨어**를 구현하여 **기본적인 보호 장치**를 마련합니다.具具具具具具具具具具 (For example, implement a FastAPI dependency or middleware that checks for a valid token in the request header for each API call. Initially, this can be a simple static token or a stub that accepts a preset key, given that integrating a full authentication service is beyond the current scope. This will at least prevent unauthorized or accidental access by requiring the presence of a known token for critical endpoints.)

또한 앞으로 **관리자 계정**과 **일반 사용자 계정**을 구분하고, 관리자 전용 페이지나 API (예: `/api/config/*`, `/api/report` 등 민감 기능)에 대해서는 **관리자 권한만** 사용할 수 있도록 하는 설계를 염두에 두고자 합니다. 이를 위해 코드 구조를 역할 기반 권한체제로 확장할 준비를 합니다 (예: JWT 토큰에 role(claim)을 포함하거나, 관리자 UI는 별도로 분리하여 보호). 동시에, 운영 환경에서는 **HTTPS**를 적용하여 **데이터 전송 암호화**를 보장하고, 개발 모드에서도 필요한 경우를 대비해 CORS 허용 도메인 설정을 점검합니다.

이러한 보안 개선 작업은 **우선순위는 낮지만 꾸준히 진행해야 할 항목**입니다. 특히 **토큰 인증**은 현재 단계에서 바로 완비하기보다도, 향후 외부 인증 시스템(예: OAuth나 사내 SSO)과 연동할 수 있도록 **미리 인터페이스를 맞춰놓는 수준**으로 구현해 둡니다. 금일 내로 간단한 토큰 체크 미들웨어를 삽입하는 것을 목표로 하며, 이는 이후 기능 개발에 큰 지장은 주지 않으면서도 기본적인 보안을 강화하는 효과가 있습니다. 장기적으로는 이 토큰 인증을 실제 사용자 관리 시스템과 연결하고, **IP 제한**이나 **VPN을 통한 관리자페이지 접근** 등 추가적인 보호 조치도 검토할 예정입니다.

장기 개선 과제 (항목 9)

장기적인 발전 방향으로 제시된 과제들은 현 단계에서는 구현되지 않지만, 프로젝트의 **미래 로드맵**으로 고려되고 있습니다. **기능목표 문서**에서 언급된 바와 같이, 퀄리티널 시스템을 한 단계 발전시키기 위한 아이디어들은 다음과 같습니다:

- **키워드 역사 큐레이션** - 특정 키워드에 대한 오랜 기간의 자료를 추적하고 분류하여, 해당 키워드의 **역사적 흐름**을 보여주는 기능입니다. 이를 위해 과거 아카이브 데이터를 체계적으로 수집/저장하고, 시간순 연대표나 출처 유형별로 묶어 볼 수 있는 **새로운 포맷의 뉴스 페이지**를 구성하는 방안을 추후 모색합니다. (예: 한 키워드에 대해 과거부터 최신까지의 관련 기사/논문/커뮤니티 글을 연대기적으로 정리하여 제공)
- **트렌드 분석 자동화** - 수집된 데이터에 기반하여 **키워드 관련 트렌드**를 자동으로 파악하고 시각화하는 기능입니다. 예를 들어, 시간에 따른 기사량 증감, 주요 이슈 변화, 커뮤니티 반응 추이 등을 분석하여 편집자나 독자에게 인사이트를 주는 것입니다. 이를 위해 Elasticsearch/OpenSearch 도입이나 통계 분석 모듈을 고려할 수 있습니다.
- **AI 요약 및 번역 고도화** - 현재는 규칙 기반의 요약과 상용 API를 통한 번역을 사용하지만, 장기적으로 **고도화된 AI 모델**을 활용하여 더 정교한 **자동 요약**과 **다국어 번역**을 제공할 수 있습니다. 예컨대, 키워드 관련 방대한 자

료를 학습한 **언어 모델**을 도입해 맥락을 파악한 요약을 생성하거나, 사용자에게 **요약본의 하이라이트**를 제공하는 기능 등을 생각해볼 수 있습니다. 또한 다국어 지원을 위해 사용자 인터페이스 및 보고서를 **국제화(i18n)**하여 영문판 쉼리저널 등을 운영할 가능성도 열어둡니다.

이러한 장기 개선 사항들은 **중기 목표 범위(2~4주)**에는 포함되지 않으므로, 당장 구현할 필요는 없습니다. 그러나 프로젝트가 안정화되고 기본 기능이 완비된 이후, **6개월 내지 1년** 범위의 로드맵에서 순차적으로 검토할 예정입니다. 현재 단계에서는 아이디어 차원으로 문서에 기록만 해두고, 추후 여건이 될 때 구체적인 기획을 통해 실행에 옮기도록 합니다.

以上과 같이, **쉼리저널 관리자 시스템**의 중기 개선 과제들을 정리하고 앞으로의 실행 방안을 확인하였습니다. 이 인수인계 보고서를 통해 새로운 채팅 세션에서도 이전 작업 맥락을 잃지 않고, **연속적으로 개발 업무를 진행**할 수 있을 것입니다. 우선순위에 따라 게이트 슬라이더와 KPI 갱신 등의 기본 개선을 완료했고, 이제 **보고서 생성 및 요약/내보내기 기능 구현**을 착수하게 됩니다. 이후에도 테스트 작성과 문서화, 보안 강화 작업을 차질 없이 수행하여, 계획된 4주차까지 모든 중기 목표를 달성하도록 하겠습니다. 향후 추가적인 요구사항이나 개선 아이디어가 나오면 이 문서를 업데이트하며, 프로젝트 완성도와 안정성을 높이는 데 지속적으로 활용하길 기대합니다. 이어지는 새 채팅창에서 우선 **항목 3 관련 기능 구현 작업**을 진행하도록 하겠습니다.
