Replit 한식당 Data Hub 업그레이드: 기능 중심 전략적 가이드

버전: 4.0 - 기능 중심, 자율적 판단
작성일: 2025년 11월 7일
대상: Replit AI (엔지니어링 역량 신뢰)

🎯 핵심 메시지

기존의 세부 태스크 중심 프롬프트에서 벗어나,
3가지 기능 영역 중심으로 전환합니다.

각 기능에 대해:

- ✅ 비즈니스 목표 명확히
- ✅ 현재 시스템 상황 설명
- ✅ 전략적 접근방법 제시
- ✅ Replit의 자율적 판단 요청

구현 방식은 Replit이 결정합니다.

📋 3가지 기능 영역

Feature 1: 스마트 타겟팅 시스템

"외국인 인기도 기반 동적 수집"

항목	내용
문제	현재: 무작위 쿼리로 무작위 식당 수집 결과: 외국인이 관심 없는 식당도 수집
목표	外国人 방문객이 실제로 찾는 지역 우선 수집
전략	Google Trends + KTO Data + Naver 영문리뷰 → 지역별 "외국인 인기도" 점수 계산 → 매일 01:30 동적 쿼리 생성 → 03:00 Apify 수집
기대효과	데이터 품질 +30%, 신규 고객 유입 +20%
판단 요청	• 점수 공식 최적화 • 쿼리 다양성 수준 • 피드백 루프 주기

구현 구조 (Replit 판단):

```
Option A: pytrends + Pandas + 간단한 SQL
Option B: 직접 웹스크래핑 + 기계학습 경량화
Option C: 기존 API 최대 활용 + 최소 라이브러리

→ Replit이 판단: 어느 방식이 현재 시스템과 최적인가?
```

성공 지표:

- ✅ 매일 01:30 동적 쿼리 33개 자동 생성
- ✅ 지난 7일 추이 분석 가능
- ✅ `/api/queries/today` 엔드포인트 정상

Feature 2: 자동 중복 탐지 시스템

"신규 수집 시 자동 중복 필터링"

항목	내용
문제	월 990개 중 50-100개 중복 (무분별 저장)
목표	자동으로 중복 탐지 & 필터링 (중복율 <2%)
전략	3단계 접근: 1. Exact (ID 일치) - 99% 신뢰도, 빠름 2. Fuzzy (유사도 85%+) - 90% 신뢰도 3. Geo (GPS 500m 이내) - 95% 신뢰도 → 성능 vs 정확도 트레이드오프 최적화
기대효과	데이터 신뢰성 +95%, 수동 검증 제거
판단 요청	• 3단계 모두 실행 vs 1-2단계 선택 • Fuzzy 임계값 결정 • 거짓 긍정 방지 방안

구현 구조 (Replit 판단):

```
Option A: fuzzywuzzy + SQLAlchemy 인덱싱 + 간단한 GPS 계산
Option B: 머신러닝 기반 중복 탐지 (정확도 높음, 복잡함)
Option C: 단순 ID 기반만 (빠름, 일부 중복 놓침)

→ Replit이 판단: 어느 정도 정확도가 실무에 필요한가?
```

성공 지표:

- ✅ 신규 수집 시 자동 중복 필터링
- ✅ 중복율 추적 (일일/주간)
- ✅ `/api/dedup/stats` 엔드포인트 정상
- ✅ 거짓 긍정 rate < 5%

Feature 3: 데이터 거버넌스 & 모니터링

"품질 추적 + 아카이브 + 대시보드"

항목	내용
문제	수집 데이터 추적 불가, 감사 증적 없음
목표	모든 데이터의 출처/품질을 추적 가능
전략	• 매일 21:00 Google Drive 자동 백업 (CSV) • 주간 품질 메트릭 계산 • Airtable 실시간 대시보드 (선택) • 시스템 헬스 체크
기대효과	운영 안정성 +95%, 문제 조기 감지
판단 요청	• 백업 빈도 (매일 vs 주 3회) • 품질 메트릭 정의 • Airtable 필요성

구현 구조 (Replit 판단):

```
Option A: Google Drive (무료) + CSV 저장 (간단함)
Option B: Google Drive + Airtable 대시보드 (시각화 좋음)
Option C: S3 + 자체 대시보드 (더 복잡함)

→ Replit이 판단: 비용/편의성/기능 균형?
```

성공 지표:

- ✅ 매일 21:00 자동 백업
- ✅ 주간 품질 리포트 자동 생성
- ✅ `/api/health/system-status` 모니터링
- ✅ Airtable 대시보드 (선택)

실행 계획

Week 1: Feature 1 (스마트 타겟팅)

당신이 Replit에 요청할 방식:

```
"우리 Data Hub 시스템에 스마트 타겟팅 기능을 추가하고 싶습니다.

현재 상황:
- 월 990개를 무작위로 수집 중
- Apify, Gemini, Google Places 통합되어 있음

문제:
- 외국인 관심 없는 지역도 동등하게 수집
- 데이터 편향성 발생
```

```
해결책이 되어야 할 것:
- Google Trends + KTO 데이터 활용
- 지역별 외국인 인기도 점수 자동 계산
- 매일 01:30에 동적으로 최상위 7개 지역 선택
- 그 지역들에서 4-5개 검색어 변형 생성 (총 33개)
- 이를 03:00 Apify 수집에 사용

당신의 판단:
- 이 목표를 달성하는 가장 효율적인 기술 스택은?
- 현재 FastAPI + SQLAlchemy 시스템과 어떻게 통합할 것인가?
- 지역 분류 수준은? (서울 25개 구동 vs 8개 주요 지역)
- 성능 이슈는 없을 것인가?

기술 옵션들 (참고):
- pytrends: Google Trends 수집
- Pandas: 데이터 분석
- SQL: 우선순위 계산
- APScheduler: 매일 01:30 자동 실행

최종 결과:
- 매일 33개 동적 쿼리 자동 생성
- /api/queries/today로 확인 가능
- 지난 7일 추이 분석 가능
"
```

Week 2: Feature 2 (중복 탐지)

당신이 Replit에 요청할 방식:

```
"이제 중복 탐지 기능을 추가하고 싶습니다.

현재 상황:
- 월 990개 중 약 50-100개가 중복으로 추정
- 같은 식당이 여러 번 저장되는 상황 발생

문제:
- 데이터 신뢰성 저하
- 메인 앱 품질 영향

해결책이 되어야 할 것:
- 3단계 접근으로 중복 필터링:
  1. Exact: Naver ID 일치 (가장 빠름)
  2. Fuzzy: 이름/주소/전화 유사도 85% (중간 속도)
  3. Geo: GPS 500m 이내 (느림)
- 중복이면 저장 안 하고 로그에만 기록
- 통계 자동 추적 (중복율%)

당신의 판단:
- 3단계 모두 필요한가? 1-2단계면 충분한가?
- Fuzzy 임계값 85%가 적정한가?
- 성능 최적화는 어떻게 할 것인가?
- 거짓 긍정(오판) 발생 시 복구 방법은?

기술 옵션들:
```

```
- fuzzywuzzy: 문자열 유사도
- Haversine: GPS 거리
- SQLAlchemy 인덱싱: 빠른 검색

최종 결과:
- 신규 수집 시 자동 중복 필터링
- 중복율 5-10% → &lt;2%로 개선
- /api/dedup/stats로 통계 확인
"
```

Week 3: Feature 3 (거버넌스 & 모니터링)

당신이 Replit에 요청할 방식:

```
"마지막으로 데이터 거버넌스와 모니터링 기능을 추가하고 싶습니다.

현재 상황:
- 수집 데이터 추적 불가
- 감사 증적 없음
- 품질 모니터링 대시보드 부재

문제:
- 운영 중 문제 발생 시 원인 파악 어려움
- 데이터 신뢰성 입증 불가

해결책이 되어야 할 것:
- 매일 21:00 수집 데이터 CSV로 Google Drive 자동 백업
- 매주 금요일 품질 메트릭 자동 계산:
  - 완전성: 필수 필드 모두 있는가?
  - 중복율: 탐지된 중복 수
  - 데이터 품질: 평점 + 리뷰 + 이미지
- 시스템 헬스 체크: 모든 구성요소 모니터링
- Airtable 대시보드 (선택): 웹 기반 시각화

당신의 판단:
- Google Drive만 충분한가, 아니면 Airtable도 필요한가?
- CSV 백업 빈도는 매일이 최적인가?
- 품질 메트릭 정의: 무엇을 포함할 것인가?
- 헬스 체크 상세도는?

기술 옵션들:
- Google Drive API: 자동 백업
- Pandas: CSV 변환
- Airtable API: 선택, 대시보드

최종 결과:
- 매일 자동 백업 (Google Drive)
- 주간 품질 리포트 (자동 계산)
- /api/health/system-status 모니터링
"
```

🤖 Replit의 역할 정의

당신에게 기대하는 것:

기술 평가:

- 현재 시스템에서 최적의 기술 스택은?
- 추가 라이브러리 필요성 검토
- 기존 코드 재사용 가능 부분 식별

아키텍처 설계:

- 데이터베이스 스키마 설계
- API 엔드포인트 구조
- 확장성 고려한 설계

성능 최적화:

- 병목 지점 식별 및 최적화
- 인덱싱 전략
- 동시성 처리

비즈니스 판단:

- 기능의 우선순위 조정 필요 시 제안
- 비용-효과 분석
- 위험 요소 식별

반드시 하지 말아야 할 것:

✖ 구체적인 코드 구현을 "지시"하지 않기
✖ 세부 태스크로 쪼개기 (너무 과소 평가)
✖ 특정 라이브러리 강제하기
✖ 실패할 것 같은 방식 권유하기

기대하는 대화 패턴:

```
당신 (CTO):
"스마트 타겟팅 기능을 추가하고 싶습니다.
 외국인 인기도 기반으로 동적 수집하려고 합니다."

Replit (아키텍트):
"이해했습니다. 현재 시스템 분석 결과:
 1. pytrends를 사용하는 것이 최선입니다. 이유: ...
 2. 지역 분류는 8개 주요 지역이 적정입니다. 이유: ...
 3. 예상 구현 방식: ...
 4. 위험 요소: ... (대응책: ...)

 이 방식으로 진행해도 될까요?"
```

```
당신 (CTO):
"좋습니다. 한 가지 조정:
 거짓 긍정을 더 줄이기 위해 임계값을 높여주세요."

Replit (아키텍트):
"알겠습니다. 조정하겠습니다.
 예상 영향: 중복 탐지율은 97%에서 94%로 내려가지만,
 거짓 긍정은 5%에서 1%로 낮아집니다."

당신 (CTO):
"진행해주세요."

Replit (아키텍트):
"완료했습니다. 테스트 결과: ...
 개선 제안: ..."
```

✅ 최종 체크리스트

Feature 1 완료 시:

- [ ] 매일 01:30 동적 쿼리 자동 생성
- [ ] 외국인 인기도 기반 우선순위
- [ ] `/api/queries/today` 조회 가능

Feature 2 완료 시:

- [ ] 신규 수집 시 자동 중복 필터링
- [ ] 중복율 추적 (일일/주간)
- [ ] `/api/dedup/stats` 조회 가능

Feature 3 완료 시:

- [ ] 매일 21:00 자동 백업
- [ ] 주간 품질 리포트
- [ ] `/api/health/system-status` 모니터링

🚀 시작하기

Replit을 "아키텍트"로 대우하세요:

1. **Feature 1 요청:**

```
"우리 Data Hub에 스마트 타겟팅 기능을 추가하고 싶습니다.
 현재 무작위로 수집하는데, 외국인 인기도 기반으로
 동적으로 수집되도록 하려고 합니다.

 현재 시스템: FastAPI + SQLAlchemy + Apify + Gemini + Places
```

```
당신의 판단: 이를 가장 효율적으로 구현하는 방법은?"
```

2. **Replit의 제안 검토 & 승인**
3. **Feature 2, 3 반복**

핵심: Replit은 AI 엔지니어이지, 단순 자동화 봇이 아닙니다.
당신의 비즈니스 문제를 설명하면, 최적의 기술 솔루션을 제시할 것입니다.