요구사항 정의서

분류	중분류	요구사항 ID	요구사항명	요구사항 상세 설명	기능 / 비기능	우선 순위	비고	변경 사항
웹 페이지 (WEB)	성과 대시보드	WEB-DAS-01	닉네임 표시	유튜브와 닉네임의 정보를 대시보드에 출력한다.	기능	하		
		WEB-DAS-02	프로필 이미지 표시	유튜브와 프로필 이미지의 정보를 대시보드에 출력한다.	기능	하		
		WEB-DAS-03	일일 방문자 수 표시	유튜브의 일일 방문자 수를 막대그래프로 시각화하여 대시보드에 출력한다.	기능	상		
		WEB-DAS-04	총 조회수 표시	유튜브 영상의 좋아요 수의 합을 대시보드에 출력한다.	기능	중		
		WEB-DAS-05	총 좋아요수 표시	시청한 영상의 목록을 대시보드에 출력한다.	기능	중		
		WEB-DAS-06	총 댓글 수 표시	총 댓글 수를 대시보드에 출력한다.	기능	중		
		WEB-DAS-07	구독자 수 표시	총 구독자 수를 대시보드에 출력한다.	기능	상		
		WEB-DAS-08	총 방문자 수 표시	총 방문자 수를 대시보드에 출력한다.	기능	상		
	영상 정보 대시보드	WEB-VDO-01	유튜브 최다 조회수 영상 공유	유튜브 영상 중 조회수가 가장 높은 영상을 공유한다.	기능	중		
		WEB-VDO-02	영상별 조회수 표시	영상별 조회수를 보여준다.	기능	상		
		WEB-VDO-03	영상별 댓글 수 표시	영상별 댓글 수를 보여준다.	기능	상		
		WEB-VDO-04	영상별 좋아요 수 표시	영상별 좋아요 수를 보여준다.	기능	상		
		WEB-VDO-05	영상별 모델 설정 정보 표시	영상별 사용한 프롬프트 및 모델 파라미터를 간략화해서 보여준다.	기능	상		
	통계 데이터 저장	WEB-STA-01	시청 통계 저장	영상별 조회수, 댓글 수, 좋아요 수 계산한 후 DB에 저장한다.	기능	상	DB	
		WEB-STA-02	채널 통계 저장	방문자 수와 구독자 수를 DB에 저장한다.	기능	상	DB	
	밈 발견	MEM-DIS-01	나무위키 밈 목록 크롤링	나무위키 '인터넷 밈/연도별 목록' 페이지에서 밈 목록을 크롤링한다. [크롤링 대상] - URL: namu.wiki/w/인터넷 밈/연도별 목록/{year}년 - 추출 범위: 월별 섹션 (1월~12월) [파싱 규칙] - 리스트 항목( ) 내 링크 텍스트 추출 - 괄호 내 파생 밈 분리: '밈A(파생1, 파생2)' → ['밈A', '파생1', '파생2'] - 빈 항목, 중복 항목 제거	기능	상	BeautifulSoup 사용	
		MEM-DIS-02	위키 노이즈 필터링	크롤링된 밈 이름에서 유효하지 않은 항목을 필터링한다. [필터링 규칙] - 길이 제한: 2자 미만 또는 80자 초과 제외 - 각주 패턴: '[숫자]' 포함 시 제외 - 목차 패턴: '숫자.' 로 시작 시 제외 - 메타텍스트: '유형하기 시작한', '항목 참조', '문서의 r숫자판' 등 포함 시 제외 - 설명문: 마침표로 끝나고 30자 초과 시 제외	기능	중	_is_valid_meme_name()	
	기본정보 수집	MEM-BAS-01	밈 정의/기원 수집	나무위키, Google 검색을 통해 밈의 정의와 기원 정보를 수집한다. [수집 항목] - definition: 밈에 대한 상세 정의 (3-5문장) - origin.source: 출처 (예: 유튜브, 트위터) - origin.creator: 제작자/원작자 - origin.date: 시작 시점 - origin.platform: 최초 유행 플랫폼 - keywords: 검색 키워드 배열 [도구] - search_namuwiki: 나무위키 검색 - search_google: Google 검색 (Serper API)	기능	상	GPT-4o 활용	
		MEM-BAS-02	위험 정보 수집	밈의 논란, 부정적 뉴스, 민감 이슈 등 위험 정보를 수집한다. [수집 항목] - risk_level: low / medium / high - controversies: 논란 사항 배열 - sensitive_topics: 민감 주제 배열 - news_sentiment: 뉴스 감정 분석 결과 [판단 기준] - high: 법적 문제, 혐오 표현, 정치적 논란 - medium: 일부 부정적 뉴스, 세대 갈등 - low: 특이사항 없음 [검색 키워드] '밈이름 + 논란', '밈이름 + 문제', '밈이름 + 비판'	기능	상	Stage 2.5 Safety 검증에 활용	
		MEM-EXA-01	활용 예시 검색	블로그에서 밈 활용 예시 URL을 검색한다. [검색 대상] - 블로그 우선 (blog.naver.com, tistory.com) - 금지: 커뮤니티 이미지 (ruliweb, dcinside, dogdrip) [검색 쿼리] '밈이름 + 활용', '밈이름 + 사용법', '밈이름 + 예시'	기능	중	Serper API	

	활용예시 수집	MEM-EXA-02	활용 예시 크롤링	검색된 URL에서 실제 활용 예시를 크롤링하여 추출한다. [필수 출력 필드] - source_url: 크롤링한 원본 URL (필수) - source_title: 원본 페이지 제목 - context: 사용 상황/맥락 - description: 활용 방법 설명 (원문 기반, 3-5문장) - original_quote: 원문에서 직접 발췌한 문장 (필수) - example_type: 챌린지 / 패러디 / 상황극 / 짤 / 댓글 [할루시네이션 방지] - 크롤링된 원문에서만 추출 - 소스에 없는 내용은 생성 금지	기능	상	3-5개 수집	
	Youtube 수집	MEM-YT-01	YouTube 쇼츠 검색	밈 관련 YouTube Shorts를 검색하여 상위 N개를 수집한다. [검색 조건] - 쿼리: 밈 이름 (meme_name 그대로 사용) - 필터: videoDuration=short (60초 이하) - 정렬: viewCount (조회수 내림차순) - 개수: 상위 3개 [수집 항목] - video_id: YouTube 영상 ID - video_title: 영상 제목 - youtube_url: 전체 URL - view_count: 조회수 [선정 기준] - 조회수 1위 영상을 video_analysis 대상으로 선정	기능	상	YouTube Data API v3	-
	영상 분석	MEM-VID-01	Gemini 영상 분석	Gemini API로 YouTube 영상을 분석하여 밈 구간과 동작을 추출한다. [입력] - video_id: 분석할 YouTube 영상 ID - meme_name: 밈 이름 - key_phrase: (선택) 탐지할 핵심 대사 [출력 - time_stamp] - start: 밈 구간 시작 시간 (초) - end: 밈 구간 종료 시간 (초) - detected_text: 탐지된 텍스트 {original, korean, language} [출력 - movement_analysis] - main_actions: 주요 동작 배열 - energy_level: low / medium / high - tempo_feel: slow / medium / fast - mood: 분위기 (cheerful, dramatic 등) - key_poses: 핵심 포즈 배열	기능	상	gemini-2.0-flash 사용	-
		MEM-VID-02	Motion Prompt 생성	Gemini가 영상을 보고 I2V 모델용 motion_prompt를 생성한다. [출력 구조] - motion_prompt_hint: 전체 동작 설명 (영문) 예: 'Cute cartoon character tilts head, curious expression, says phrase with playful tone' - motion_sequence: 시간대별 동작 배열 예: [{time: '0-0.5s', action: 'Head tilts right'}, ...] - body_parts: 신체 부위별 움직임 예: {head: 'tilts right', face: 'curious expression', arms: 'still'} - style_keywords: 스타일 키워드 배열 예: ['cute', 'kawaii', 'anime', 'smooth animation'] - negative_prompt: 피해야 할 요소 기본값: 'static, stiff, realistic, blurry, distorted' - duration_seconds: 영상 길이 (기본 5초) - camera_motion: static / pan / zoom	기능	상	Wan2.2-I2V 입력용	-
	밈 유형 분류	MEM-TYP-01	영상 기반 1차 분류	영상 분석 결과에서 detected_text 유무로 1차 분류한다. [QUOTABLE (대사형)] - 조건: detected_text 존재 + 텍스트 길이 > 2 - 처리: 즉시 확정 → key_phrase = detected_text.korean 또는 detected_text.original → needs_audio = true [PERFORMABLE (동작형) 후보] - 조건: detected_text 없음 또는 빈 문자열 - 처리: MEM-TYP-03 (LLM 분류)로 진행	기능	상	meme_typing_node()	-
		MEM-TYP-02	동작 키워드 탐지	동작 관련 키워드로 PERFORMABLE 여부를 판단한다. [동작 키워드] - '댄스', '춤', '챌린지', 'dance', 'challenge' - movement_analysis.main_actions 참조 [판단 규칙] - 키워드 2개 이상 → PERFORMABLE 확정 - 키워드 1개 이하 → MEM-TYP-03 (LLM 분류)로 진행	기능	중	meme_typing_node()	-

밈 데이터 관리 (MEM)		MEM-TYP-03	LLM 기반 보조 분류	detected_text 없는 경우 LLM으로 분류한다. [입력] - meme_name: 밈 이름 - definition: 밈 정의 - motion_prompt_hint: 영상 분석 결과 - detected_text: (빈 문자열) [LLM 프롬프트 판단 기준] - QUOTABLE: 대사/발음이 밈의 핵심 - PERFORMABLE: 동작/춤이 밈의 핵심 [출력] - meme_type: quotable 또는 performable - key_phrase: 핵심 대사 (quotable인 경우) - reason: 판단 근거 [기본값] - LLM 실패 시 → PERFORMABLE	기능	중	GPT-4o-mini 사용	-
	음성 분석	MEM-AUD-01	밈 구간 오디오 추출	밈 구간의 오디오를 WAV 파일로 추출한다. [실행 조건] - meme_type = 'quotable' - time_stamp.start, time_stamp.end 존재 [처리 과정] 1. YouTube 영상 다운로드 (yt-dlp) 2. FFmpeg로 해당 구간 오디오 추출 3. WAV 형식 (16kHz, mono) 저장 [출력] - 파일 경로: data/raw/audio/{video_id}_{start}_{end}_{timestamp}.wav - 파일 크기: 약 500KB (3초 기준) [캐싱] - 영상 캐시: data/raw/video_cache/{video_id}.mp4 - 중복 다운로드 방지	기능	상	quotable만 실행	-

		MEM-AUD-02	운율(Prosody) 분석	librosa로 추출된 오디오의 운율 특성을 분석한다.
				
				[분석 항목]
				- pitch: 평균/최소/최대 피치 (Hz)
				- pitch_variation: 피치 변동폭
				- tempo: 말속도 (syllables/sec 추정)
				- energy: RMS 에너지 프로파일
				- duration: 실제 발화 길이
				
				[출력 구조]
				prosody: {
				pitch_mean: 220.5,
				pitch_range: [180, 350],
				tempo: 'fast',
				energy_pattern: 'rising',
				speaking_rate: 4.2
				}
					기능	상	librosa 사용	quotable만

		MEM-AUD-03	SSML 동적 생성	GPT-4o-mini로 TTS용 SSML 마크업을 동적 생성한다. [입력] - key_phrase: 핵심 대사 - prosody: 운율 분석 결과 - meme_context: 밈 맥락 정보 [출력 예시] <speak> <prosody rate='fast' pitch='+10%'> <emphasis level='strong'>나니가</emphasis> <break time='100ms'/> 스키? </prosody> </speak> [SSML 요소 활용] - prosody: rate, pitch, volume 조절 - emphasis: 강조 구간 - break: 쉼표, 끊어읽기 - phoneme: 발음 힌트	기능	중	Voice Clone 힌트	quotable만
		MEM-SEL-01	Google Trends 점수 조회	pytrends로 밈의 최근 트렌드 점수를 조회한다. [조회 조건] - 기간: 최근 7일 (now 7-d) - 지역: 한국 (geo='KR') - 언어: 한국어 (hl='ko') [점수 계산] - Google Trends 상대 점수 (0-100) - 7일 평균값 사용 - API 실패 시 기본값: 50 [배치 처리] - 5개씩 배치 조회 (API 제한) - 배치 간 1초 대기	기능	상	pytrends 라이브러리	-
		MEM-SEL-02	Safety 검증	risk_info 기반으로 위험한 밈을 제외한다. [필터링 규칙] - risk_level = 'high' → 제외 (rejection_reason: 'high_risk') - risk_level = 'medium' 또는 'low' → 통과 [추가 키워드 검증] - controversies 키워드: '법적', '소송', '형오' 포함 시 제외 - sensitive_topics 키워드: '정치', '종교' 포함 시 제외	기능	상	_filter_by_safety()	-

	밈 선정	MEM-SEL-03	Freshness 검증	최근 사용된 밈을 제외하여 콘텐츠 다양성을 확보한다. [검증 규칙] - meme_usage_history 테이블 조회 - 최근 14일 내 사용 기록 있음 → 제외 - 사용 기록 없음 → 통과 [신선도 점수 계산] - 마지막 사용일 기준 경과일 계산 - 공식: min(100, (days_since / 30) × 100) - 30일 이상 미사용: 100점 - 15일 미사용: 50점 - 사용 이력 없음: 100점	기능	중	14일 쿨다운	-
		MEM-SEL-04	우선순위 점수 계산	가중치 기반으로 최종 우선순위 점수를 계산한다. [점수 공식] total_score = (trend_score × 0.6) + (freshness_score × 0.4) [가중치] - WEIGHT_TREND: 0.6 (60%) - WEIGHT_FRESHNESS: 0.4 (40%) [선정] - total_score 내림차순 정렬 - 상위 N개 선정 (기본 5개)	기능	상	상위 N개 선정	-
	에러처리	MEM-ERR-01	Rate Limit 재시도	API Rate Limit 발생 시 지수 백오프로 재시도한다. [재시도 설정] - 최대 재시도: 3회 - 기본 대기: 60초 - 지수 백오프: 60초 → 120초 → 240초 [대상 API] - Gemini API (429 RESOURCE_EXHAUSTED) - OpenAI API (429 에러) - Serper API (Rate Limit) [구현] - retry_on_rate_limit() 데코레이터 - invoke_with_retry() 함수	비기능	상	모든 외부 API	-
		MEM-ERR-02	부분 실패 허용	개별 밈 처리 실패 시 전체 파이프라인은 계속 진행한다. [에러 처리] - 개별 밈 예외 → errors 배열에 기록 - 다음 밈 처리 계속 - 최종 결과에 성공/실패 개수 포함 [성공 판정 기준] - meme_id 존재 (DB 저장 완료) 또는 - definition 존재 (데이터 수집 완료) [실패 시 기록] - meme_name: 실패한 밈 이름 - error: 에러 메시지 - phase: 실패한 Phase	비기능	상	-	-
	캐싱	MEM-CAC-01	영상/오디오 캐싱	다운로드된 영상과 추출된 오디오를 로컬에 캐싱한다. [영상 캐시] - 경로: data/raw/video_cache/{video_id}.mp4 - 중복 다운로드 방지 - 캐시 히트 시 '[CACHE]' 로그 [오디오 캐시] - 경로: data/raw/audio/{video_id}_{start}_{end}_{timestamp}.wav - 동일 구간 재추출 방지	비기능	중	디스크 공간 관리 필요	-
	로깅	MEM-LOG-01	단계별 로깅	각 Phase의 시작/완료 시간과 결과를 로깅한다. [로그 형식] - 시작: '[Phase: {name}] Started - meme: {meme_name}' - 완료: '[Phase: {name}] Completed ({duration}s)' - 에러: '[Phase: {name}] Error: {message}' [로그 레벨] - INFO: 정상 처리 - WARNING: 비정상 but 계속 진행 - ERROR: 실패 [로거] - Python 표준 logging 모듈 - AgentLogger 클래스 (scripts/utils/logging.py) - 파일 로깅: setup_file_logging()	비기능	중	표준 logging	-
	AI 에이전트 영상 생성	GEN-AGT-01	시나리오 골격 생성	밈 데이터를 바탕으로 5단 구조 시나리오 골격을 생성한다. 이때 생성된 시나리오 골격은 다음 단계인 GEN-AGT-02~03에서 활용된다. - 입력: 밈 데이터 - 출력: 5단 구조로 시나리오 생성 (hook → setup → buildup → meme → punchline) 이때 대사는 [PLACEHOLDER: 설명] 형태로 남김	기능	상		
		GEN-AGT-02	대사 생성	골격의 placeholder를 캐릭터별 실제 대사로 변환한다. 1. 파인튜닝한 텍스트 생성 모델을 통해 일관된 페르소나를 유지 (부장, 사원) 2. 자연스러운 대화체 유지	기능	상		
		GEN-AGT-03	최종 시나리오 생성	생성된 시나리오 골격과 행동/대사를 정리하여 전달한다. 1. 캐릭터 음성 모델로 정리된 대사를 전달해야 한다. 1-1. 이때 대사에서 밈이 사용되고, 그 밈이 영상 매체의 대사나 노래 가사로부터 기원하였을 경우, 음성 모델에서 음성 파일을 활용할 수 하도록 지시를 해야 한다. 2. 캐릭터 이미지 모델로 배경 이미지와 캐릭터의 행동을 정리하여 전달해야 한다. 3. 영상 생성 모델에게 캐릭터의 행동과 배경의 변화를 전달해야 한다. 4. 영상으로 제작될 콘텐츠의 제목과 설명, 자막, 태그를 생성해야 한다.	기능	상		
		GEN-AGT-04	캐릭터 음성 모델	GEN-AGT-03에서 생성한 캐릭터의 대사와 음성 설명을 바탕으로 음성을 생성한다. 이때, 음성 생성 모델은 파인튜닝한 모델이다. 1. 일관된 목소리와 말투를 유지해야 한다. 2. 음성에 감정이 반영되어야 한다. 3. 캐릭터 대사에서 활용된 밈이 영상 매체의 대사나 노래 가사로부터 기원하였을 경우, 그 피치와 박자를 유사하게 따라해야 한다.	기능	상		

콘텐츠 생성 (GEN)		GEN-AGT-05	씬별 이미지 생성	각 씬별 상황 배경과 캐릭터 기준 이미지를 토대로 이미지로 생성한다. 1. 일관된 캐릭터 외형을 유지해야 한다. (이미지 임베딩을 통한 유사도 비교나 LLM을 통한 평가) 2. 이미지의 분위기나 그림체를 유지해야 한다. 3. 제공될 플랫폼에 맞는 화면 비율로 생성해야 한다.	기능	상		
		GEN-AGT-06	씬별 영상 생성 모델	이미지와 음성 파일을 바탕으로 영상을 생성한다. 1. 9:16의 비율로 생성해야한다. 2. 프롬프트로 지시사항을 입력할 수 있어야 한다. 3. 주어진 음성에 맞는 립싱크가 되어야 한다. 4. 영상의 개별 프레임끼리 이미지 유사도가 높도록 제작되어야 한다.	기능	상		
		GEN-AGT-07	최종 편집	씬별로 생성된 영상과 음성을 FFmpeg를 통해서 통합하여 최종 영상을 생성한다.	기능	상		
	콘텐츠 평가	GEN-EVA-01	자체 평가 시스템	모델들을 통해 생성된 것을 자체적으로 평가할 수 있는 로직을 구성한다.	기능	상		
		GEN-EVA-02	영상 피드백 루프	영상별 좋아요 수, 댓글 수, 구독자 증가 수와 영상 생성 시 사용된 프롬프트와 로직을 매칭시킨다.	기능	중		
		GEN-EVA-03	이전 각본 컨텍스트 반영	이전 영상 각본을 불러와 세계관 및 스토리 흐름을 유지한다.	기능	중		
		GEN-EVA-04	영상-음성 싱크	영상과 음성의 싱크가 맞아야 한다.	기능	중		
콘텐츠 업로드 (DIS)	유튜브 Shorts 자동 업로드	DIS-YSU-01	YouTube Data API 연동	OAuth2 인증을 통해 채널 연결 및 API 호출 환경을 구성한다.	기능	상		
		DIS-YSU-02	자동 업로드 수행	렌더링 완료된 영상을 YouTube Shorts 형식으로 자동 업로드한다.	기능	상		
		DIS-YSU-03	실패 처리 및 재시도	업로드 실패 시 재시도 로직을 수행하고 결과를 기록한다.	기능	상		
	인스타그램 릴스 자동 업로드	DIS-IRU-01	Instagram Graph API 연동	Business/Creator 계정 인증 및 API 연결을 수행한다.	기능	상		
		DIS-IRU-02	릴스 자동 업로드	영상을 릴스 형식으로 자동 업로드한다.	기능	상		
		DIS-IRU-03	실패 처리 및 재시도	업로드 실패 시 재시도 로직을 수행하고 결과를 기록한다.	기능	상		
	업로드 성과 모니터링	DIS-DPM-01	업로드 상태 추적	각 플랫폼 업로드 성공·실패·대기 상태를 추적한다.	기능	상		
		DIS-DPM-02	성과 데이터 집계	수집된 조회·시청·반응 이벤트를 집계한다.	기능	중		
		DIS-DPM-03	성과 데이터 분석 및 저장	수집 데이터를 저장하고 분석에 활용 가능하도록 제공한다.	기능	중		
		DIS-DPM-04	예약 업로드 스케줄링	과거 업로드 성과 데이터(조회수, 반응률)를 기반으로 최적 업로드 시간을 분석해 자동 예약 업로드를 설정한다.	기능	중		
		DIS-DPM-05	알림 및 경고 관리	파이프라인 전 단계에서 발생한 실패·에러를 관리자에게 알림으로 전달한다.	기능	상		
	저장 및 배포	DIS-STR-01	콘텐츠 저장 및 메타데이터 기록	생성 완료된 콘텐츠를 Media Storage에 업로드하고 저장 경로를 반환받아 메타데이터를 DB에 기록한 후 저장 완료 상태를 Backend에 전달한다.	기능	중		
모델 학습 (MDL)	파인튜닝 학습 데이터 수집	MDL-DAT-01	스크립트 생성 모델용 원천 데이터 수집	스크립트 생성 모델 파인튜닝 학습을 위한 원천 대화 데이터를 수집한다. - 데이터 출처: KoCulture-Dialogues(HuggingFace), Office 상황 데이터셋(AI Hub) - 데이터 유형: 실제 일상 대화 - 저장 형식: TSV / JSONL	기능	상		
		MDL-DAT-02	캐릭터 기반 대화 데이터 생성	수집된 원천 대화 데이터를 참고하여 캐릭터의 성격과 말투 기준에 맞는 파인튜닝용 대화 다이얼로그 데이터를 생성한다. - 생성 방식: GPT-5.1 API 활용 - 데이터 규모: 캐릭터 당 약 5,000개 - 데이터 형식: JSONL	기능	상		
		MDL-DAT-03	학습 데이터 전처리 및 검수	생성된 대화 데이터를 파인튜닝 학습에 적합하도록 전처리한다. - 캐릭터와 일치하는 상황 지문(감정, 행동) 추가 - 불필요한 발화 제거 및 포맷 통일 - 학습 스키마에 맞게 필드 정규화	비기능	상		
		MDL-DAT-04	음성 데이터셋 구축	캐릭터별 목소리 특성에 맞는 음성 데이터를 수집하여 파인튜닝용 데이터셋을 구축한다. - Alhub에서 수집한 데이터로 vocal_path|speaker_name|language|text 형식의 데이터셋을 구성한다. - 원천 음성 데이터 (wav) - 라벨링 데이터 (json)	기능	상		
		MDL-DAT-05	음성 학습 데이터 전처리	- 수집된 음성 및 라벨 데이터를 TTS 학습에 적합한 형식으로 변환한다. - 전처리 단계: - 샘플링 레이트 통일 (예: 32kHz) - 음성 볼륨 정규화 - 잡음 제거 및 불필요한 구간 삭제 - 음성 길이 및 텍스트 길이 검증 - 텍스트 전처리: 문장 부호 통일, 불필요 공백 제거, 한글/영문 인코딩 검수	기능	상		
		MDL-DAT-06	음성 학습 데이터 검수	- 수집된 음성 및 라벨 데이터를 품질을 검수한다. - 검수 단계: - 음성과 텍스트 매칭 확인 - 발음 오류, 잡음 등 음질 확인 - 라벨링(json)과 음성 파일 경로(vocal_path) 일치 여부 확인	비기능	상		
	스크립트 생성 모델 파인튜닝	MDL-SLT-01	스크립트 생성 모델 파인튜닝	사전 학습된 언어 모델을 기반으로 숏폼(60초 이내) 영상에 적합한 스크립트 생성을 목표로 파인튜닝을 수행한다. - 출력 대상: 상황과 지문을 기반으로 한 캐릭터 특성에 맞는 대사	기능	상		
		MDL-SLT-02	페르소나 및 톤앤매너 제약 학습	스크립트 생성 시 사전에 정의된 페르소나와 말투 스타일을 일관되게 유지하도록 학습 제약 조건을 적용한다. - 캐릭터 성격 일관성 유지 - 말투 및 어휘 스타일 고정	기능	상		
		MDL-SLT-03	스크립트 생성 품질 평가 기준 수립	파인튜닝된 모델이 생성한 스크립트의 품질을 평가하기 위한 기준을 수립하고 테스트를 통해 성능을 검증한다. - 평가 항목: - 각본 길이 적합성 - 스토리 흐름 및 일관성 - 톤앤매너 일관성	비기능	중	YanoljaNEXT-EEVE-Instruct-10.8B	
	TTS 음성 모델 파인튜닝	MDL-TTS-01	음성 합성 모델 학습	GPT-SoVITS S2 캐릭터 음성 파인튜닝 - 목적: 쇼츠 등장인물 캐릭터의 고유 음색과 발화 스타일을 유지하며 일관성 있는 음성을 생성하기 위함. - 설명: 사전 학습된 TTS 모델(GPT-SoVITS S2)을 기반으로 특정 캐릭터의 음성 특성과 발화 패턴에 맞춰 파인튜닝을 수행한다.	기능	상		
		MDL-TTS-02	감정 표현 학습	- 목적: 대본 내 문장 태그(감조, 유머, 진지 등)에 따른 감정 표현을 정확히 모델링하기 위함. - 설명: 학습 데이터에 포함된 문장 태그를 기반으로 모델이 음성 톤, 발화 속도, 강세 등을 조절하도록 학습한다.	기능	상		
		MDL-TTS-03	음성 품질 평가	- 목적: 학습된 모델의 음성 생성 품질을 객관적/주관적으로 평가하고, 파인튜닝 성능을 검증하기 위함. - 설명: 생성된 음성을 다양한 지표로 평가하여 모델의 자연스러움, 발음 정확도, 감정 표현 적합도를 확인한다.	비기능	중		
	파인튜닝 제어	MDL-CMN-01	파인튜닝 버전 관리	데이터셋, 하이퍼파라미터, 학습 결과를 버전 단위로 관리하여 모델 성능 저하 시 변경이력과 영향 요소를 추적할 수 있게 한다.	기능	중		
		MDL-CMN-02	모델 롤백 지원	품질 저하 발생 시 이전 안정 버전 모델로 즉시 복구 가능하게 한다.	기능	상		
데이터 수집 파이프라인		ANL-DCP-01	콘텐츠 배포 및 반응 수집 준비	완성된 콘텐츠를 플랫폼 정보와 트래킹 ID를 설정하여 반응 분석 모듈에 전달하고 초기 성과 데이터 수집을 시작한다.	기능	중		

성과 분석 (ANL)	데이터 수집 파이프라인	ANL-DCP-02	멀티 플랫폼 데이터 통합	유튜브, 인스타그램의 성과 데이터(조회수, 좋아요, 댓글, 공유 등)를 API를 통해 통합 저장한다.	기능	중		
	성과 지표 분석	ANL-PMA-01	공유율 분석	조회수 대비 공유 비율을 계산하여 바이럴 잠재력이 높은 콘텐츠 특성(예: 특정 밈 카테고리, 각본 스타일)을 식별한다.	기능	중		
		ANL-PMA-02	플랫폼별 성과 비교	동일 콘텐츠의 유튜브, 인스타그램 성과를 비교한다.	기능	중		
	예측 분석 및 인사이트	ANL-PAI-01	최적 업로드 시간 추천	플랫폼별, 요일별, 시간대별 성과 데이터를 분석하여 조회수와 반응률이 가장 높은 최적 업로드 시간을 자동으로 추천한다.	기능	중		
비기능 (NFR)	성능	NFR-PER-01	영상 로딩 응답 시간	사용자가 피드에서 영상을 선택하거나 자동 재생 시, 영상 초기 로딩은 2초 이내에 시작되게 한다. 단, 네트워크 환경에 따른 예외 상황은 허용한다.	비기능	상		
	안정성	NFR-REL-01	크롤링 실패 허용	특정 플랫폼 크롤링 실패 시에도 전체 수집 파이프라인은 중단되지 않게 한다.	비기능	중		
		NFR-REL-02	업로드 실패 복구	업로드 실패 발생 시 재시도 로직을 수행하며, 실패 이력은 로그로 기록한다.	비기능	중		
	보안	NFR-SEC-01	인증 정보 보호	OAuth 토큰 및 API Key는 서버 환경 변수 또는 암호화 저장소에 저장한다.	비기능	상		
		NFR-SEC-02	악성 입력 방어	댓글 및 검색 입력값에 대해 XSS, SQL Injection을 방지하기 위한 입력 검증 및 필터링을 수행한다.	비기능	상		
	개인정보 보호	NFR-PRV-01	개인정보 최소 수집	개인 식별이 가능한 정보는 수집하지 않거나 최소한으로 제한한다.	비기능	중		
		NFR-PRV-02	사용자 행동 데이터 익명화	시청 기록 및 반응 데이터는 개인을 식별할 수 없도록 익명화하여 분석에 활용한다.	비기능	중		
	유지보수성	NFR-MNT-01	로그 및 모니터링	크롤링, 업로드, 추천 등 지속적 모니터링이 필요한 기능은 로그 수집 및 모니터링 대상이 되게 하며, 장애 원인 분석 및 성능 개선에 활용한다.	비기능	중		
		NFR-MNT-02	설정 기반 운영	수집 주기, 추천 가중치 등의 주요 운영 파라미터는 코드 수정 없이 설정값으로 조정 가능하게 한다.	비기능	하		
	확장성	NFR-SCL-01	데이터 수집 확장성	밈 크롤링 소스는 향후 플랫폼 추가가 가능하도록 모듈화된 구조로 설계되게 한다. 각 플랫폼 크롤러는 독립적으로 추가·제거 가능하게 하며, 기존 수집 파이프라인에 영향을 주지 않게 한다.	비기능	하		
domain level								