

FalseCam

최종보고서

팀명
주석열
팀장
서민석
팀원
김주현, 최태열
개발주제
"AI 기반 이미지 생성 기술을 활용한 감성 브이로그 콘텐츠 생성 웹 애플리케이션 - FalseCam"

목차

제1장 개요	4
1.1 목적 및 기대효과	4
1.1.1. 목적	4
1.1.2. 기대효과	4
1.2 프로젝트 개요	4
1.2.1. 배경	4
1.2.2. 사용 대상자	4
1.2.3. 범위 및 기능(역할)	4
1.3 관련 기술	5
1.4 업무분장 및 일정계획표	6
1.5 원가 분석	7
1.5.1. 인건비	7
1.5.2. 인건비 이외 항목	7
1.5.3. 판매가	7
제2장 분석 공정	8
2.1 수요 분석	8
2.2 업무 흐름도	8
2.2.1. 사용자	8
2.2.2. 관리자	8
2.3 프로세스 분석	9
2.3.1. Context Diagram	9
2.3.2. 자료 흐름도(Data Flow Diagram)	10
2.3.3. 기능 명세서	11
2.4 인터페이스 분석	15
2.4.1. 입·출력 인터페이스 종류와 목적	15
2.4.2. 입·출력 인터페이스 내용	15
제3장 설계 공정	16
3.1 시스템 구성	16
3.1.1. 시스템 환경	16
3.2 자료 설계	17
3.2.1. 자료 사전	17
3.2.2. 테이블 구성	17
3.3 코드 설계	18
3.3.1. 구조 차트(Structure Chart)	18
3.3.2. 모듈 개요도	19

3.3.3. 모듈 설명서	19
3.3.4. 엔티티-프로세스 매트릭스	25
3.4 핵심 기능	26
3.5 인터페이스 설계	30
 제4장 구현 공정	 35
4.1 프로그램 작성	35
4.1.1. 코딩 컨벤션 (Back)	35
4.1.2. 코딩 컨벤션 (Front)	35
4.2 커밋 메시지 컨벤션	36
4.2.1. 기본구조	36
4.2.2. 타입 (Type)	36
4.2.3. 스코프 (Scope)	36
4.2.4. 제목 (Subject)	36
4.2.5. 커밋메시지 예시	37
4.3 디자인 시스템 및 UI 가이드	37
4.3.1. 디자인 원칙	37
4.3.2. 색상 팔레트	37
4.3.3. 타이포그래피	38
4.3.4. 레이아웃	39
4.3.5. 핵심 컴포넌트	39
4.4 API 컨벤션	39
4.4.1. 기본 URL	39
4.4.2. 일반 규칙	39
4.4.3. 표준 응답 구조	39
4.4.4. API 명세	40
 제5장 배포	 43
5.1. Cloudflare Pages 설정	43
5.2. Render 설정	45
 제6장 결 론	 47
6.1 프로젝트 성과 및 의의	47
6.2 향후 발전 방향	48
6.3 맺음말	49

제1장 개요

1.1 목적 및 기대효과

1.1.1. 목적

- 사용자의 평범한 일상을 텍스트와 사진으로 입력받아 특별한 경험으로 재구성하는 AI 서비스 제공
- 복잡한 기술이나 전문 지식 없이 누구나 쉽게 자신만의 이미지와 영상을 생성
- 간단한 한글 입력만으로 AI가 최적의 프롬프트를 자동 생성하여 사용자 편의성 극대화
- 생성형 AI 기술의 진입 장벽을 낮추어 콘텐츠 제작의 대중화에 기여

1.1.2. 기대효과

1) 기술적 파급효과

- LLM을 활용한 '이야기 기반 자동 프롬프트 생성' 기술은 AI의 창의적 활용 모델을 제시
- 텍스트, 이미지, 비디오 AI를 순차적으로 연동하는 파이프라인 구조는 향후 다양한 AI 서비스 개발에 응용 가능

2) 사회·산업적 파급효과

- 복잡한 절차 없이 누구나 영상 콘텐츠를 제작할 수 있어 콘텐츠 산업의 저변을 확대
- 전통적인 기록 방식을 '디지털 저널리즘'으로 확장시켜 새로운 창작 및 감성 경험을 제공

3) 사용자 경험 확장

- 자신의 사진과 글로 디지털 페르소나를 만들며 새로운 방식으로 자신을 표현하는 재미를 제공
- SNS 프로필, 블로그 등 개인 온라인 공간을 독창적인 콘텐츠로 꾸밀 수 있음
- AI 기술을 직접 체험하며 '생성형 AI'에 대한 이해를 자연스럽게 높일 수 있는 기회 제공

1.2 프로젝트 개요

1.2.1. 배경

- 생성형 AI 기술이 빠르게 발전함에 따라 사용자별로 특색있는 디지털 콘텐츠 제작에 대한 관심이 증가했습니다.
- 기존의 필터나 효과를 넘어, 각자의 이야기가 담긴 개성 있는 콘텐츠에 대한 필요성이 대두되었습니다.
- 이에 따라 복잡한 기술 지식 없이도 누구나 자신의 일상을 창의적인 이미지와 영상으로 재구성할 수 있는 서비스가 필요하다고 판단하여 본 프로젝트를 기획했습니다.

1.2.2. 사용 대상자

- 평범한 일상을 특별한 콘텐츠로 기록하고 싶은 모든 사용자
- SNS 프로필, 블로그 등 자신만의 온라인 공간을 개성 있게 꾸미고 싶은 사용자
- 전문적인 기술 지식 없이 쉽고 재미있게 창작 활동을 경험하고 싶은 사용자
- 생성형 AI 기술을 가볍고 직관적인 방식으로 체험해보고 싶은 사용자

1.2.3. 범위 및 기능(역할)

1) 핵심 기능

- **AI 이미지 생성:** 사용자가 입력한 일상 텍스트와 사진을 기반으로 AI 이미지를 생성합니다. 백엔드에서는 입력된 텍스트를 LLM(GPT-3.5-Pro)을 통해 분석하고, 이미지 생성에 최적화된 영어 프롬프트로 변환합니다. 이 프롬프트와 사용자 이미지를 Flux-Kontext-Pro모델에 전달하여 최종 이미지를 생성하고 결과물 URL을 반환합니다
- **AI 비디오 생성:** 이미지 생성이 완료되면, 해당 이미지 URL과 프롬프트를 활용해 즉시 Kling-2.1-standard모델을 호출하여 짧은 비디오를 생성합니다
- **결과물 조회 및 관리:** 생성된 이미지와 영상은 썸네일 리스트로 표시되며, 클릭 시 모달 창에서 좌우로 넘겨보며 확인할 수 있습니다. 사용자는 원하는 결과물을 다운로드할 수 있습니다.

2) 사용자 관리

- **익명 세션 기반:** 별도의 회원가입 절차 없이 서비스를 이용할 수 있습니다.
- **세션 ID 활용:** 사용자가 처음 접속할 때 고유 세션 ID(SSID)를 생성하여 로컬 스토리지에 저장합니다. 모든 API 요청 시 이 ID를 HTTP 헤더에 담아 전송하여 사용자를 식별합니다.
- **서버 역할:** 백엔드 서버는 헤더로 전달된 세션 ID를 기준으로 요청을 처리하고, 생성된 결과물을 세션별 폴더에 분리하여 저장 및 관리합니다.

3) 범위

- OS windows 11 기준 PC환경

1.3 관련 기술

1) 프론트엔드 (Frontend)

- React, Vite, JavaScript, Tailwind CSS

2) 백엔드 (Backend)

- Python, FastAPI, Uvicorn

3) AI 모델 (AI Models)

- LLM: GPT-3.5-Pro
- Text-to-Image: Flux-Kontext-Pro
- Image-to-Video: Kling-2.1-standard

4) 배포 (Deployment)

- Frontend: Cloudflare Pages
- Backend: Render

1.4 업무분장 및 일정계획표

구 분			기간	2025년 7월 ~ 2025 8월				
업무명(모듈명)			담당자	2 주	3 주	4 주	5 주	1 주
1.0 착수 및 기획	1.1 기획 및 설계	WBS 수립	서민석	→				
		화면 설계	최태열	→				
		스택 확정	김주현	→				
	1.2 프로젝트 관리	기획서 작성	서민석	→				
		Git 전략 수립	공통	→				
2.0 MVP 구 현 및 검 증	2.1 개발 환경설정	Cloudflare 연동	김주현		→			
		React 설정	최태열		→			
		Model Testing	서민석		→			
	2.2 백엔드개발	이미지 생성 API 개발	김주현		→			
		백엔드 서버 배포	서민석		→			
	2.3 프론트개발	UI 컴포넌트 개발	최태열		→			
		메인페이지 연동	최태열		→			
		비디오 생성 호출	최태열		→			
	2.4 통합 및 테스트	프로덕션 환경 구축	공통		→			
		MVP 테스트	공통		→			
3.0 고도화 및 마무리	3.1 백엔드개발	비디오 생성 API 개발	김주현			→		
	3.2 프론트개발	session_id 적용	최태열			→		
		생성 결과 표시 강화	최태열			→		
	3.3 최종검수	기능 통합	공통			→		
		프로덕션 배포	공통				→	
		최종 보고서 완성	서민석				→	

1.5 원가 분석

1.5.1. 인건비

항목	관련정보	비고
프로젝트 관련 인원	3명	팀원 총 3명
개발 인원 시급	10,030원 / 시간	참여인원 모두가 취업자가 아니기 때문에 2025년 최저임금 적용
일 평균 근로시간	10.4시간	2025년 법정 최대 근로시간 (주 52시간) / 5일
총 프로젝트 제작 기간 (2025.07.08.~ 2025.07.23.)	12일	주말 및 공휴일 제외, 기획서 작성 기간 포함
인건비 총 예상 금액	3,755,232 원	$3명 \times 10,030원 \times$ $10.4시간 \times 12일$ $= 3,755,232 원$

1.5.2. 인건비 이외 항목

- 장비 및 S/W 비용은 학원에서 제공하는 PC와 프로그램을 사용하여 별도의 금액이 들지 않았다. 소모품 및 회의비용은 학원 시설을 이용하여 추가적인 비용은 발생되지 않았다.

1.5.3. 판매가

- 금액: 유료
- 본 서비스는 외부 AI API(예: 이미지·영상 생성 등) 사용료와 클라우드 서버 운영비가 주요 원가로 포함된다. AI API 1회 호출당 약 400원의 비용이 발생하며, 하루 100회 사용 시 일일 약 40,000원, 월 1,200,000원(30일 기준), 연간 약 **14,600,000원**(365일 기준)의 API 사용료가 소요된다.

제2장 분석 공정

2.1 수요 분석

1) 주 타겟층:

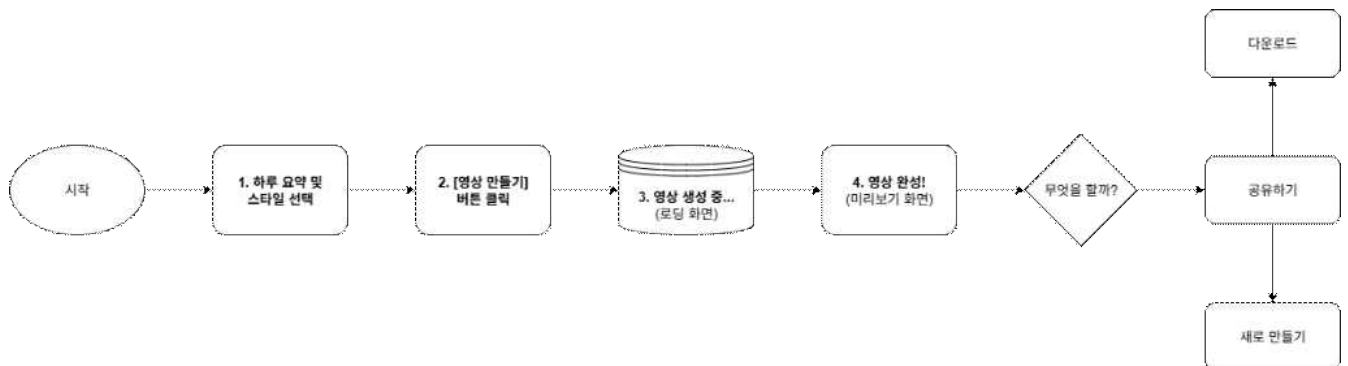
- 자신의 평범한 일상을 특별한 것으로 만들고 싶어 하는 주변의 평범한 사람들.

2) 수요 분석:

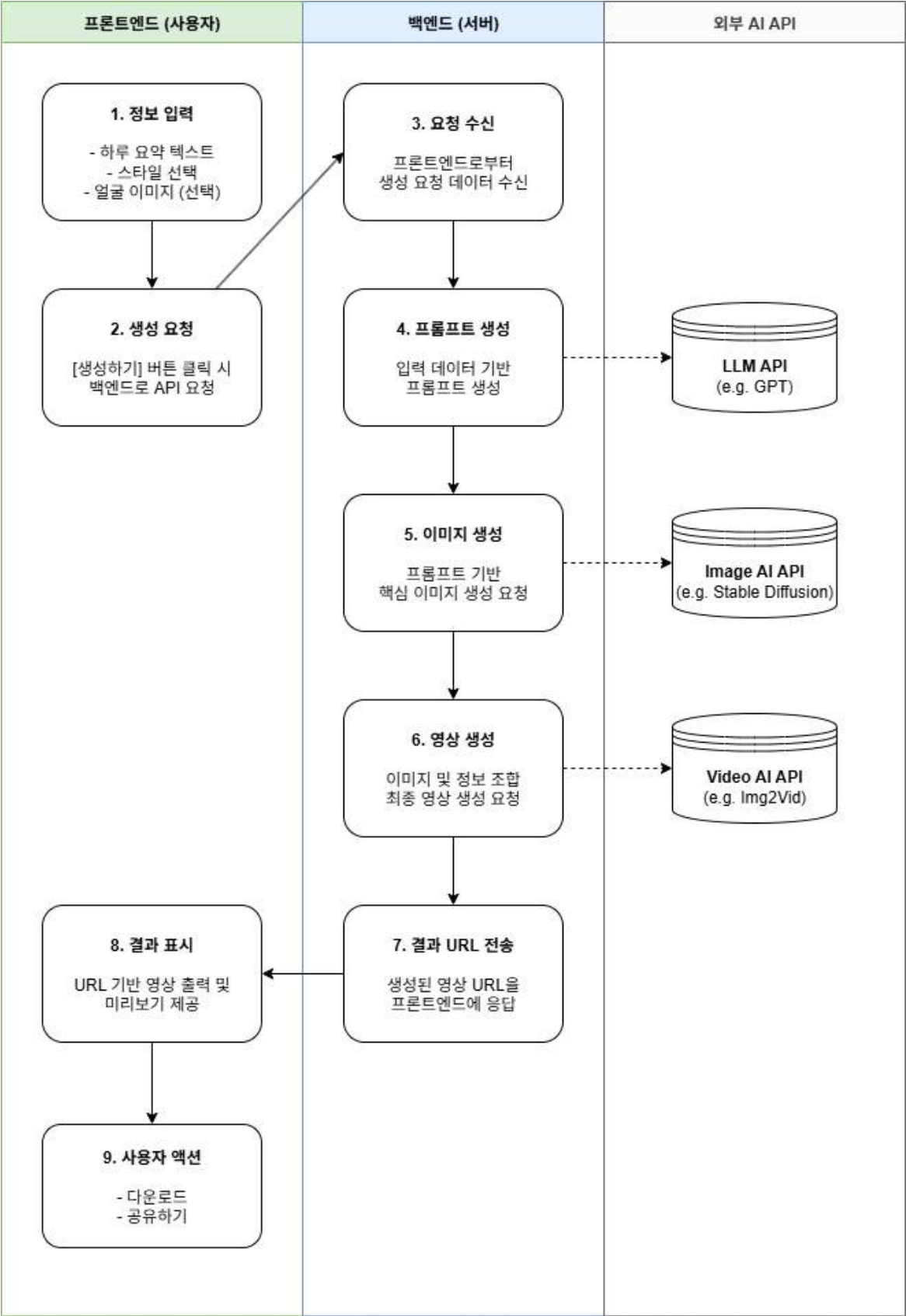
- 즉각적인 재미와 만족감: 단순한 문장 하나로 상상도 못한 이야기가 생성되길 바램. 과정은 단순하되 결과는 빠름.
- 창의적인 영감과 놀이의 재료: 어떤 텍스트를 넣으면 더 재미있는 결과가 나올지 궁금증 유발 어디까지 만들어내는지 구경하고 싶어 할 것.
- 정서적 위안과 대리만족: 우리가 기획한 시나리오처럼, 평범한 하루 끝에 FalseCam을 통해 현실에 없는 경험(세계 여행, 패러글라이딩 등)을 간접적으로 체험하며 위안을 얻음

2.2 업무 흐름도

2.2.1. 사용자

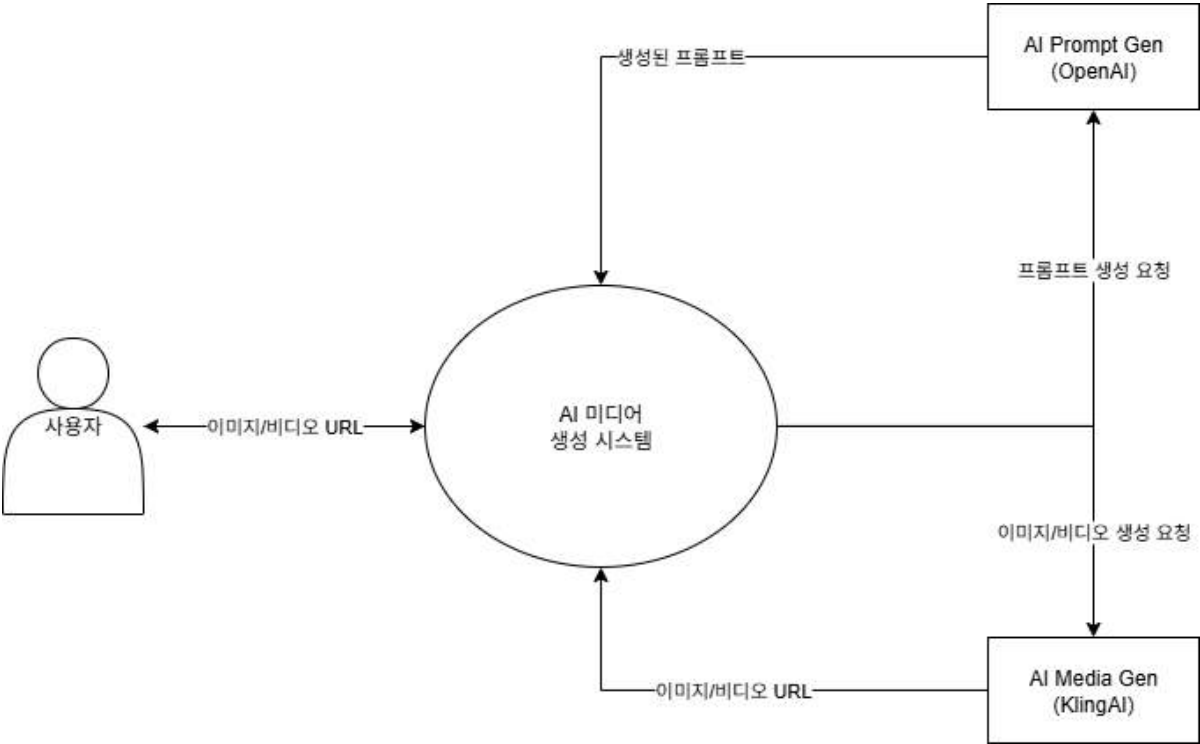


2.2.2. 관리자

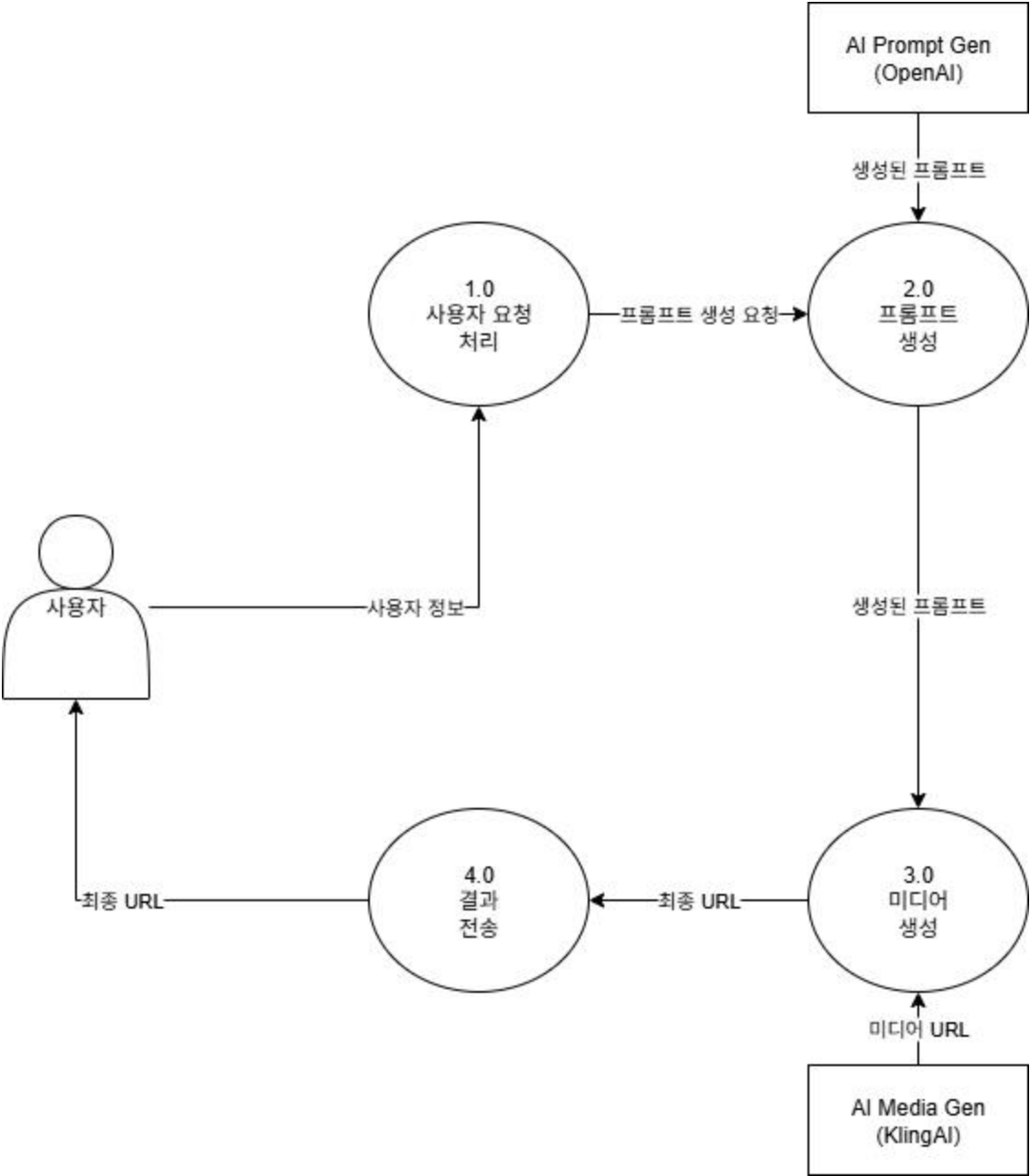


2.3 프로세스 분석

2.3.1. Context Diagram



2.3.2. 자료 흐름도(Data Flow Diagram)



2.3.3. 기능 명세서

기능명	AI 이미지 및 비디오 생성
기능 정의	사용자가 입력한 텍스트와 사진을 기반으로 AI가 개인화된 이미지와 비디오를 생성
관련 엔티티	익명 세션(Anonymous Session) 데이터
입력 데이터	<ul style="list-style-type: none"> - 사용자의 텍스트 - 사용자의 사진 파일 - 개인화 옵션 (스타일, 나이, 성별)
출력 데이터	<ul style="list-style-type: none"> - 생성된 이미지 파일(PNG) URL - 생성된 동영상 파일(MP4) URL
상세 설명	
<ol style="list-style-type: none"> 1. 사용자가 메인 화면에서 텍스트를 입력하고, 사진을 업로드한 뒤 원하는 스타일의 옵션을 선택합니다. 2. '오늘 하루 소환' 버튼을 클릭하면 프론트엔드는 모든 입력 데이터와 세션 ID를 백엔드 서버로 전송합니다. 3. 백엔드 서버는 AI 모델들을 순차적으로 호출하여 텍스트를 프롬프트로 변환하고, 이를 바탕으로 이미지와 비디오를 생성합니다. 4. 생성된 결과물의 URL이 프론트엔드로 반환되고, 사용자는 화면에서 이미지와 비디오를 확인 및 다운로드할 수 있습니다. 	
요구사항 및 제약조건	
<ul style="list-style-type: none"> - 텍스트 입력 필드가 비어있을 경우, 사용자에게 입력을 요청하는 안내를 표시해야 합니다. - 이미지나 비디오가 생성되는 동안, 사용자에게 진행 중임을 알리는 로딩 상태(애니메이션, 진행바 등)를 명확히 표시해야 합니다. - AI 서버에서 오류가 발생할 경우, 사용자에게 생성에 실패했음을 알리고 다시 시도하도록 안내해야 합니다. - 외부 서버에 저장된 결과물을 다운로드할 때, 브라우저 보안 정책(CORS)을 우회하기 위해 파일 데이터를 Blob 객체로 변환하여 다운로드시키는 기능이 구현되어야 합니다. 	

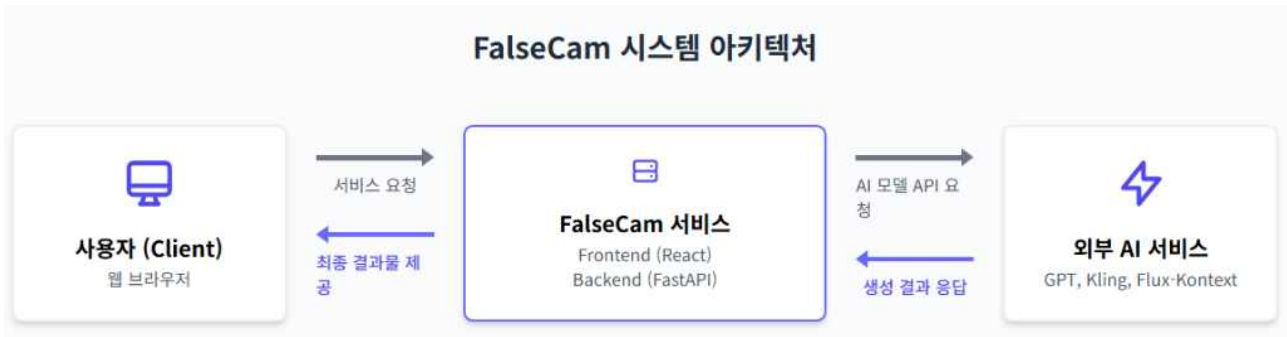
기능명	생성 결과물 다운로드 (Result Download)
기능 정의	외부 서버에 저장된 AI 생성 이미지 또는 비디오를 사용자의 로컬 기기에 저장
관련 엔티티	이미지/비디오 파일 URL
입력 데이터	다운로드할 콘텐츠의 URL
출력 데이터	이미지(PNG/JPG) 또는 비디오(MP4) 파일
상세 설명	
<ol style="list-style-type: none"> 1. 사용자가 결과물 조회 화면에서 '다운로드' 버튼을 클릭합니다. 2. fetch API를 사용해 외부 서버에 있는 파일 데이터를 가져와 Blob객체로 변환합니다 . 3. URL.createObjectURL()을 이용해 해당 Blob에 대한 임시 로컬 URL을 생성합니다 . 4. 보이지 않는 <a>태그를 동적으로 생성하고, href속성에 임시 URL을, download속성에 파일명을 지정합니다. 5. 생성된 <a>태그에 대해 click()이벤트를 실행하여 브라우저의 다운로드 기능을 트리거합니다 . 6. 다운로드 실행 후, 메모리 누수를 방지하기 위해 URL.revokeObjectURL()로 임시 URL을 폐기합니다 . 	
요구사항 및 제약조건	
<ul style="list-style-type: none"> - 브라우저의 보안 정책(CORS)으로 인해 외부 서버의 파일을 직접 다운로드할 수 없는 문제를 해결해야 합니다. - 이미지 파일뿐만 아니라 비디오 파일 등 다양한 형식의 파일을 처리할 수 있어야 합니다. - 임시로 생성된 객체 URL은 반드시 메모리에서 해제하여 브라우저의 성능 저하를 방지해야 합니다 . 	

기능명	AI 프롬프트 생성 (AI Prompt Generation)
기능 정의	사용자가 입력한 한국어 텍스트를 LLM(대규모 언어 모델)을 통해 이미지 생성 AI에 최적화된 상세 영문 프롬프트로 변환
관련 엔티티	시스템 프롬프트(System Prompt)
입력 데이터	사용자의 일상 텍스트 (한국어)
출력 데이터	상세하고 구체적인 영문 프롬프트
상세 설명	
<ol style="list-style-type: none"> 1. 백엔드 서버는 사용자로부터 전달받은 한국어 텍스트를 입력값으로 사용합니다 . 2. 단순 번역을 넘어, 장면의 배경, 분위기, 인물의 행동과 의상 등을 풍부하게 묘사하도록 설계된 시스템 프롬프트(system_prompt)를 미리 정의합니다 . 3. 이 시스템 프롬프트와 사용자 텍스트를 함께 OpenAI API(GPT)에 전달하여, AI가 스스로 추론하고 창의적인 장면을 구성하도록 유도합니다 . 4. API 호출 결과로 반환된, 시각적 정보가 풍부한 최종 영문 프롬프트를 다음 이미지 생성 단계로 전달합니다.. 	
요구사항 및 제약조건	
<ul style="list-style-type: none"> - 시스템 프롬프트는 최종 결과물의 품질을 결정하므로, 명확한 배경, 긍정적인 분위기, 인물의 구도 (정면 또는 3/4 측면) 등 핵심적인 요구사항이 반드시 포함되어야 합니다. - 인물의 얼굴을 묘사하지 않고 오직 장면과 행동에만 집중하도록 AI에게 명확히 지시해야 합니다. - 사용자가 입력한 단순한 상황(예: '공원 산책')을 기반으로 AI가 어울리는 의상(예: '캐주얼한 후드 티와 청바지')을 자연스럽게 제안하도록 유도해야 합니다. 	

기능명	익명 세션 관리 (Anonymous Session Management)
기능 정의	회원가입 없이 사용자를 식별하고, 연속적인 서비스 이용을 지원하며, 사용자별 데이터를 관리
관련 기능	세션 ID (SSID), 세션별 로그 및 데이터 폴더
관련 엔티티	예약 DB (reserve, reserve_time), 공유누리 Open-API
입력 데이터	없음
출력 데이터	고유 세션 ID (SSID)
상세 설명	
<ol style="list-style-type: none"> 1. 사용자가 서비스에 처음 접속하면, 프론트엔드는 로컬 스토리지에 세션 ID가 있는지 확인합니다 2. 세션 ID가 없으면 고유한 ID를 새로 생성하여 로컬 스토리지에 저장합니다 . 3. 프론트엔드는 백엔드에 API를 요청할 때마다 저장된 세션 ID를 HTTP 헤더에 담아 전송합니다 . 4. 백엔드 서버는 이 세션 ID를 기반으로 요청을 구분하고, 생성된 결과물을 세션별 폴더에 저장하며 로그를 기록합니다 . 	
요구사항 및 제약조건	
<ul style="list-style-type: none"> - 모든 API 요청에는 반드시 세션 ID가 포함되어야 연속적인 작업 흐름이 보장됩니다 . - 서버는 세션 ID를 기준으로 결과물을 분리 저장하여 사용자 데이터가 섞이지 않도록 해야 합니다 - 본 방식은 비회원 기반 서비스이므로, 사용자 인증 없이도 개별 요청을 식별하고 추적할 수 있어야 합니다 . 	

2.4 인터페이스 분석

2.4.1 입·출력 인터페이스 종류와 목적



1) 프로그램 사용자(Client):

- FalseCam 애플리케이션에 일상 텍스트, 사진 등 재료가 되는 정보를 입력하여 새로운 콘텐츠 생성을 요청합니다.

2) 서비스 제공자(FalseCam 서비스):

- 사용자의 요청을 받아 백엔드에서 외부 AI 서비스(LLM, 이미지/비디오 생성 모델)를 순차적으로 호출합니다.
- 각 AI 서비스로부터 응답받은 결과(이미지, 비디오 URL)를 조합하여 사용자에게 최종 콘텐츠를 제공합니다.
- 별도의 DB 대신, 각 사용자의 요청을 **익명 세션 ID** 기반으로 식별하고 관리합니다

2.4.2. 입·출력 인터페이스 내용

1) 이용자

- 메인 생성 인터페이스:

- 사용자가 일기를 입력하는 텍스트 필드
- AI 변환에 사용할 사진을 업로드하는 영역
- 결과물의 스타일, 나이, 성별 등을 선택하는 옵션 카드
- 이미지 및 비디오 생성을 시작하는 "오늘 하루 소환" 버튼

- 결과물 표시 영역:

- 생성된 이미지와 비디오를 보여주는 썸네일 카드 리스트
- 썸네일 클릭 시 결과물을 크게 보고 좌우로 넘길 수 있는 전체 화면 모달
- 현재 선택된 결과물을 기기에 저장하는 다운로드 버튼

- 상태 표시:

- AI가 콘텐츠를 생성하는 동안 표시되는 로딩 애니메이션 및 진행 바
- 입력 조건이 맞지 않는 등 특정 상황에 사용자에게 안내를 제공하는 팝업

제3장 설계 공정

3.1 시스템 구성

3.1.1. 시스템 환경

항목	내용
OS	WINDOWS 11
IDE	Visual Studio Code
LANGUAGE	Python 3.10, JavaScript
FRONT-END	React.js, Tailwind CSS, Vite
BACK-END	FastAPI, Uvicorn, python-multipart, requests, python-dotenv
AI API	OpenAI, fal.ai (text-to-image / image-to-image / image-to-video)
PKG	pip, npm
ETC	replicate, base64, logging, Pydantic

3.2 자료 설계

3.2.1. 자료 사전

자료이름	자료내역	자료형	필수	비고
text	생성 텍스트	STRING	Y	일기 내용 (100자 이내)
style	생성 스타일	STRING	N	realistic, 2d, dot 등 선택지 중 택 1
age	나이대	STRING	N	“20” → 25세로 매핑
gender	성별	STRING	N	male/female/other
sessionID	세션 식별자	STRING	Y	UUID 문자열, 헤더로 전달
image	참조 이미지	FILE	N	PNG,JPG 등 이미지 파일만 삽입가능, 최대 5MB

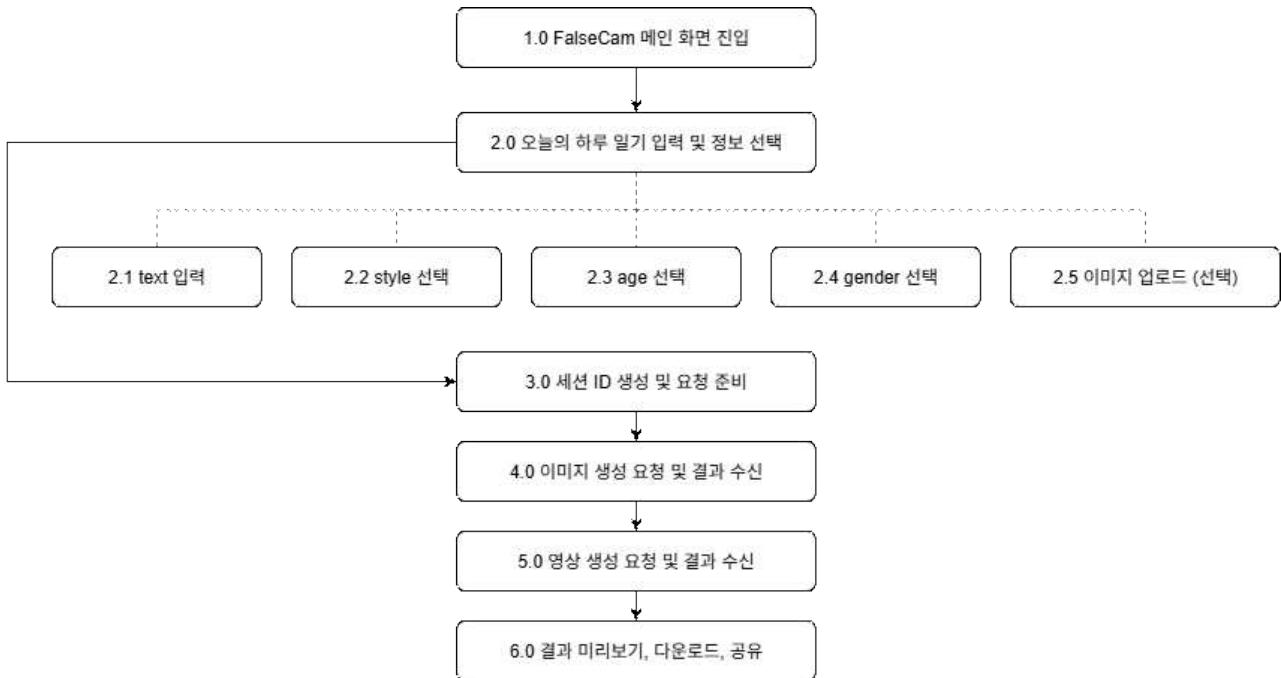
3.2.2. API 입력 데이터 명세

자료항목	자료형	자료길이	실례	코드화
session_id	STRING	36	a7a2...	UUID
text	STRING	100	오늘 카페에서 공부했어	
style	STRING	-	2d	realistic / 2d / 3d / cyberpunk / dot
age	STRING	-	20	“9”, “10”, “20” /...
gender	STRING	-	여성	male / female / other
image	FILE	-	selfie.png	jpg, png, webp 등

3.3. 코드 설계

3.3.1. 구조 차트(Structure Chart)

- 1) 이용자



3.3.2. 모듈 개요도

모듈 번호	보충 설명
1.0	FalseCam 메인 화면 진입
2.0	오늘의 하루 일기 입력 및 정보 선택(style, age, gender, image 선택삽입)
3.0	세션 ID 생성 및 요청 준비
4.0	이미지 생성 요청 및 결과 수신
5.0	영상 생성 요청 및 결과 수신
6.0	결과 미리보기, 다운로드 및 공유

3.3.3. 모듈 설명서

- 1) 이용자

1.1) FalseCam 메인 화면 진입

모듈명	FalseCam 메인 화면 진입	모듈 ID	1.0
설 명	FalseCam 서비스 메인 화면으로 진입하여 전체 생성 기능 흐름의 시작점 역할		
관련 클래스	Home.jsx, MainTitle.jsx		
입력 데이터	-		
출력 데이터	메인 UI 화면		
단 계	처 리 내 역		비 고
1	사용자가 접속 시 앱이 로딩되고 메인 화면 출력		
2	사용자의 하루 텍스트 작성, 정보 선택 등 흐름 진입할 수 있다.		

모듈명	오늘의 하루 일기 입력 및 정보 선택(style, age, gender)	모듈 ID	2.0
설 명	사용자가 하루의 일기 텍스트와 이미지 생성 스타일, 나이, 성별, 참고할 이미지를 선택하여 생성 요청에 필요한 데이터를 입력하는 단계		
관련 클래스	DailyJournalInput.jsx, InfoInputCard.jsx		
입력 데이터	text, style, age, gender, image		
출력 데이터	formData 준비 전 raw 입력값		
단 계	처 리 내 역		비 고
1	사용자가 하루 내용을 입력한다. 최대 100자		
2	생성 스타일을 선택한다.		
3	나이/성별을 선택한다.		
4	참고 이미지를 업로드한다.(선택)		
5	모든 입력값은 이후 FormData로 변환된다.		

모듈명	세션 ID 생성 및 요청 준비	모듈 ID	3.0
설 명	사용자의 생성 요청 흐름을 추적하기 위한 고유 세션 ID를 생성하고, 입력값을 FormData로 구성하여 서버 요청을 준비한다.		
관련 클래스	session.js, Home.jsx, InfoInputCard.jsx		
입력 데이터	text, style, age, gender, image		
출력 데이터	session_id, FormData 객체		
단 계	처 리 내 역		비 고
1	세션 ID, UUID로 자동 생성(최초 방문시)		
2	사용자의 입력 데이터를 기반으로 FormData 객체를 생성한다.		image 포함 시 파일 첨부
3	생성된 세션 ID는 요청 헤더에 포함된다.		key : sessionID

모듈명	이미지 생성 요청 및 결과 수신	모듈 ID	4.0
설 명	사용자 입력값을 기반으로 이미지 생성을 요청하고, 결과 이미지를 base64 데이터 URL로 수신하여 클라이언트에 전달하는 단계이다.		
관련 클래스	Home.jsx, server.py, services.py (/generate/image)		
입력 데이터	FormData, session_id		
출력 데이터	base64 인코딩된 이미지 데이터 (렌더링용), 저장된 이미지 URL (영상 생성용)		
단 계	처 리 내 역		비 고
1	FastAPI 서버의 /generate/image 엔드포인트로 POST 요청을 전송한다.		FormData + sessionID 헤더 포함
2	서버 내에서 OPENAI를 활용해 사용자의 text 입력값을 영어로 번역해 프롬프트를 생성한다.		GPT OPENAI
3	변환된 프롬프트는 fal.ai API로 전달되어 이미지를 생성한다.		text-to-img img-to-img
4	생성된 이미지를 base64 데이터 URI로 인코딩하여 클라이언트에 전달한다.		프론트로 바로 렌더링
5	영상 생성을 위해 별도로 이미지의 URL도 임시확보된다.		로컬, DB에는 저장되지않음. 페이지 이탈 시 삭제

모듈명	영상 생성 요청 및 결과 수신	모듈 ID	5.0
설 명	생성된 이미지의 URL을 기반으로 영상 생성을 요청하고, 완료된 영상을 사용자에게 전달받아 함께 출력하는 단계이다.		
관련 클래스	Home.jsx, services.py(/generate/video)		
입력 데이터	image_url, prompt, sessionID		
출력 데이터	video_url		
단 계	처 리 내 역		비 고
1	이미지 생성 후, 해당 이미지 url과 프롬프트로 영상 생성 요청을 보낸다.		POST /generate/video
2	서버에서 fal.ai API에 요청을 전송하여 영상 생성 작업을 시작한다.		img-to-video
3	생성 완료 후 영상 url을 응답으로 수신한다.		.mp4
4	프론트에서 이미지와 함께 영상도 동시 출력한다.		사용자에게 동시생성처럼 보임

모듈명	결과 미리보기, 다운로드 및 공유	모듈 ID	6.0
설 명	생성된 이미지와 영상 결과를 사용자에게 미리보기 형태로 화면에 제공하고, 다운로드 또는 sns 공유가 가능하도록 처리하는 단계이다.		
관련 클래스	ImagePreviewModal.jsx, VlogRecordCard.jsx, ActionButtons.jsx		
입력 데이터	image(base64), video_url		
출력 데이터	렌더링 된 썸네일, 다운로드 파일(mp4/png)		
단 계	처 리 내 역		비 고
1	이미지와 영상 결과가 함께 리스트에 저장된다.		imagelist 배열
2	사용자가 화면을 클릭하면 전체 미리보기 Modal이 열린다.		
3	미리보기 화면에서 다운로드 버튼 클릭 시 이미지/영상이 저장된다.		base64 →fetch 후 blob 변환
4	여러 생성 결과는 슬라이드 형식으로 넘겨볼 수 있다.		

3.3.4. 엔티티-프로세스 매트릭스

엔티티 프로세스 매트릭스

프로세스 엔티티	세션 생성 및 요청 준비	이미지 생성 요청	영상 생성 요청	결과 출력 및 다운로드
session	C	R	R	
image		C	R	R
video			C	R

C: 생성, R: 조회

엔티티 모듈 매트릭스

엔티티 모듈	SESSION	IMAGE	VIDEO
메인 화면 진입			
일기 입력 및 정보 선택			
세션 ID 생성 요청 및 결과 수신	C		
이미지 생성 요청 및 결과 수신	R	C	
영상 생성 요청 및 결과 수신	R	R	C
결과 미리보기, 다운로드, 공유		R	R

3.4 핵심 기능

- 1) 하루 일기 기반 정보 입력 기능

파일명	Home.jsx, services.py, session.js
요약	이미지/영상 생성을 위한 입력값 수집 단계. 텍스트와 선택 옵션, 이미지 업로드 포함.
<p>[Frontend]</p> <p>텍스트 입력. 최대 100자.</p> <p>스타일, 성별, 나이 옵션 선택.</p> <p>이미지 업로드 (선택). 미리보기 제공.</p> <p>입력값 상태 저장. FormData 구성.</p> <p>[설명]</p> <p>입력값 기반 생성 흐름 진입.</p> <p>라디오 방식 구성. UX 간결화.</p> <p>업로드 이미지 미리보기. blob URL 렌더링.</p> <p>백엔드 연동 준비 완료.</p>	

- 2) 이미지 생성 기능

파일명	Home.jsx, services.py
요약	사용자 입력값 기반 이미지 생성 요청. base64 인코딩 결과 렌더링.
<p>[Frontend]</p> <p>handleGenerate() 실행. FormData 구성. API로 POST 전송. 응답 수신 후 base64 데이터 상태 저장. 태그에 바인딩. 실시간 출력. sessionID 헤더 포함.</p> <p>[Backend]</p> <p>/generate/image 라우트 처리. 입력값 + 선택 이미지 기반 생성 요청. OpenAI + fal API 호출. base64 결과 JSON 반환. 예외 발생 시 500 응답. 에러 메시지 포함.</p> <p>[설명]</p> <p>핵심 생성 기능. 텍스트 → 이미지 흐름 중심. 이미지 미첨부 시에도 텍스트 생성 가능. 파일 미저장. 메모리 내 응답 처리. 세션 단위 결과 분리 관리.</p>	


- 3) 영상 생성 기능


파일명	Home.jsx, services.py
요약	생성된 이미지 URL과 프롬프트 기반으로 영상 생성 요청. 영상 URL 수신 후 렌더링.
<p>[Frontend]</p> <p>이미지 생성 완료 후 handleGenerate() 실행. image_url, prompt JSON 구성. API로 POST 전송. 응답으로 video_url 수신. imageList 상태에 저장. <video> 태그에 바인딩. 미리보기 출력. sessionId 헤더 포함.</p> <p>[Backend]</p> <p>/generate/video 라우트 처리. image_url, prompt 기반 fal API 호출. 결과에서 video.url 추출. 성공 시 JSON 응답. 실패 시 에러 반환.</p> <p>[설명]</p> <p>이미지 생성 후 자동 연결 처리. 영상 요청은 비동기 처리. 순차 전송. 영상은 CDN 또는 외부 URL 형식. 별도 저장 없음. session 기준으로 관리.</p>	

- 4) 결과 미리보기 및 저장/공유 기능

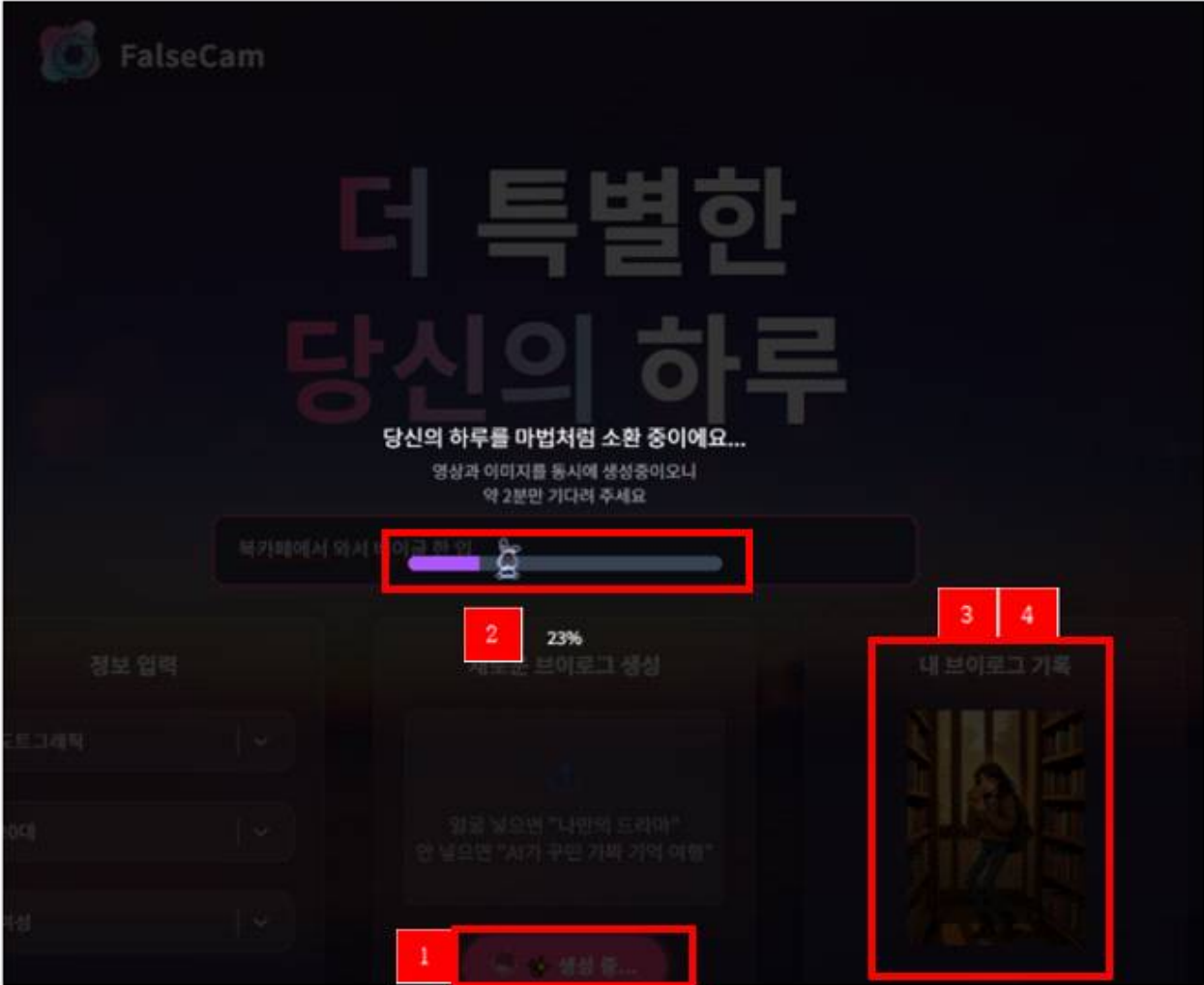
파일명	VlogRecordCard.jsx, ImagePreviewModal.jsx, ActionButtons.jsx, CustomAlertModal.jsx
요약	생성된 이미지 및 영상 결과를 카드 형태로 렌더링. 다운로드 및 SNS 공유 지원.
<p>[Frontend]</p> <p>imageList 배열 기반 결과 카드 렌더링.</p> <p>카드 클릭 시 전체 모달 미리보기 출력.</p> <p>다운로드 버튼 클릭 시 fetch → blob → 링크 다운로드.</p> <p>[설명]</p> <p>결과는 이미지/영상 타입에 따라 분기 출력.</p> <p>다운로드는 blob 객체 생성 후 자동 저장 처리.</p> <p>공유는 클립보드 복사 또는 외부 링크 연결 방식.</p> <p>image/video 모두 session 단위로 묶여 있음.</p>	

3.5. 인터페이스 설계

인터페이스명	인트로 문구 화면
설명	서비스 컨셉 소개 및 기억 입력 텍스트 작성 화면
	
이벤트	액션
1. 타이틀 문구 출력	브랜드 메시지 “더 특별한 당신의 하루” 출력
2. 예시 문구 / 사용자 문구 입력창	사용자가 기억을 100자 이내로 입력 (예: 주말 오후의 커피 등)

인터페이스명	정보 입력 화면
설명	브이로그 생성을 위한 조건(주제, 나이, 성별)을 선택하는 화면
	
이벤트	액션
1. 화풍 선택	드롭다운에서 이미지 생성화풍 선택 (예: 2d, 사이버펑크, dot)
2. 나이 선택	연령대(예: 10대, 20대, 30대 등) 선택
3. 성별 선택	“남성”, “여성”, “선택 안 함” 중 선택
4. 미선택 상태 유지	아무것도 선택하지 않으면 랜덤으로 자동 생성 문구 안내

인터페이스명	브이로그 생성 화면
설명	사용자 정보와 선택값을 바탕으로 AI가 브이로그를 생성하는 화면
이벤트	액션
1. 얼굴 사진 업로드	이미지 업로드 시 “나만의 드라마” 컨셉으로 얼굴합성 이미지 생성됨
2. 얼굴 미업로드 상태 유지	업로드하지않으면 “AI가 꾸민 가짜 기억 여행” 컨셉으로 랜덤 얼굴 이미지 생성됨
3. “오늘 하루 소환!” 버튼 클릭	정보입력창, 이미지 사진이 합쳐진 브이로그 요청이 서버로 전송되며, 업로드된 사진은 생성 후 삭제됨

인터페이스명	브이로그 생성 중 화면
설명	영상과 이미지 생성이 동시에 진행되며, 사용자에게 현재 상태를 안내하는 UI가 기존 화면 위에 오버레이되는 대기 화면
	
이벤트	액션
1. ‘오늘 하루 소환!’ 클릭	기존 화면 위에 생성 중 UI가 반투명 레이어 형태로 오버레이됨
2. 진행률 애니메이션 표시	진행 퍼센트, 텍스트, 애니메이션 바가 화면 중앙에 덧씌워져 표시됨
3. 이미지 먼저 생성 완료 시	기록 창에 실시간 썸네일로 우선 표시됨, 영상은 계속 생성 중
4. 영상 생성 완료 시	브이로그 카드가 완성되고 생성 중 레이어는 100%로 채워진 뒤 자동으로 사라짐

인터페이스명	내 브이로그 기록 화면	
설명	생성된 브이로그 이미지와 날짜를 확인하며 이전 기록을 슬라이드 방식으로 탐색할 수 있는 화면	
<div><div><div><div>내 브이로그 기록</div><div>1-1</div><div>브이로그가 아직 없습니다.</div></div></div><div><div><div>내 브이로그 기록</div><div>2-1</div><div><div><div></div></div></div><div>2-2</div><div><div>2-3</div><div>Wed, July 23, 2025</div></div></div></div></div>		
상태	이벤트	액션
1. 기록 없음	1-1.페이지 진입	“브이로그가 아직 없습니다” 문구 출력
2. 기록 있음	2-1.브이로그 생성완료	브이로그 이미지와 영상이 카드 형태로 표시됨
	2-2. 슬라이드 조작	좌/우 화살표 또는 인디케이터로 이전/다음 카드로 이동
	2-3. 날짜 확인	0하단에 생성일자 표기 (예: Wed, July 23, 2025)

제4장 구현공정

4.1 프로그램 작성

4.1.1. 코딩 컨벤션(Back)

1) 네이밍 컨벤션

- 파일/폴더: snake_case 사용 (예: services.py, __init__.py)
- 클래스: PascalCase 사용 (예: VideoRequest)
- 함수/변수: snake_case 사용 (예: generate_image, session_id)
- 상수: ALL_CAPS 사용 (예: FAL_API_KEY)

2) 프로젝트 구조

- server.py: FastAPI 앱 생성, 미들웨어 설정 등 서버의 전체적인 구성을 담당합니다.
- services.py: 실제 API 엔드포인트와 핵심 비즈니스 로직(AI 연동 등)을 포함합니다.
- .env: API 키 등 민감한 정보는 환경 변수로 분리하여 관리합니다.

3) 코드 스타일

- 들여쓰기: PEP 8 가이드라인에 따라 공백 4칸을 사용합니다.
- 최대 라인 길이: 가독성을 위해 한 줄에 80~100자를 넘지 않도록 합니다.
- 주석: 복잡한 로직이나 API 호출 부분에는 #을 사용하여 명확한 주석을 작성합니다.

4) API 엔드포인트

- 리소스 기반의 명확한 경로를 사용합니다. (예: /generate/image, /generate/video)
- URL 경로에는 동사보다 명사를 사용하고, 하이픈(-)으로 단어를 구분합니다.

4.1.2. 코딩 컨벤션(Front)

1) 네이밍 컨벤션

- 파일/컴포넌트: PascalCase 사용 (예: ImagePreviewModal.jsx, Home.jsx)
- 함수/변수: camelCase 사용 (예: handleGenerate, imageList)
- CSS 클래스: kebab-case 사용을 원칙으로 하나, Tailwind CSS의 유틸리티 클래스를 우선적으로 활용합니다.

2) 프로젝트 구조

- components: 재사용 가능한 작은 UI 단위 (버튼, 모달 등)를 관리합니다.
- pages: 라우팅의 단위가 되는 페이지 컴포넌트(예: Home.jsx)를 관리합니다.
- utils: 세션 관리(session.js)와 같이 여러 곳에서 사용되는 유틸리티 함수를 관리합니다.

3) 코드 스타일

- 들여쓰기: 공백 2칸을 사용합니다.
- 상태 관리: useState를 사용하여 컴포넌트의 상태를 명확하게 관리하고, 상태 변경 함수명은 set 접두사를 사용합니다. (예: const [isOpen, setIsOpen] = useState(false);)
- 주석: 복잡한 로직이나 이벤트 핸들러에는 // 또는 {/* */}를 사용하여 주석을 작성합니다.

4) 스타일링 (Styling)

- Tailwind CSS 우선 사용: 모든 스타일은 Tailwind CSS의 유틸리티 클래스를 사용하여 HTML 구조 내에서 직접 정의하는 것을 원칙으로 합니다
- index.css: 전역적으로 필요한 최소한의 스타일(폰트, 기본 배경색 등)만 정의합니다.
- 컴포넌트 스타일: 각 컴포넌트에 종속된 스타일은 컴포넌트 파일 내부에 작성하거나, 불가피할 경우 컴포넌트명과 동일한 CSS 파일을 만들어 관리합니다.

4.2 커밋 메시지 컨벤션

4.2.1. 기본구조

- 모든 커밋 메시지는 아래와 같은 기본 구조를 가집니다.

Type(Scope): Subject

Body (선택사항)

Footer (선택사항)

4.2.2. 타입 (Type)

- 커밋의 성격을 나타내는 타입을 제목 가장 앞에 작성합니다.

feat: 새로운 기능 추가

fix: 버그 수정

refactor: 코드 리팩토링 (기능 변경 없이 코드 구조 개선)

style: 코드 스타일 변경 (포매팅 등 기능 변경이 없는 경우)

chore: 빌드 관련 파일 수정, 패키지 매니저 설정 등 (기타 변경)

docs: 문서 수정 (README.md, 기획서 등)

4.2.3. 스코프 (Scope)

- 커밋이 영향을 미치는 코드의 영역을 명시하며, 프로젝트 맞춤형으로 아래와 같이 정의합니다.

api: 백엔드 API 로직 (services.py 등)

ui: 프론트엔드 UI 컴포넌트 (Home.jsx, Modal 등)

prompt: 프롬프트 엔지니어링 관련 로직

session: 익명 세션 관리 관련 로직

download: 파일 다운로드 기능 관련 로직

deploy: 배포(Render, Cloudflare) 관련 설정

4.2.4. 제목 (Subject)

- 커밋에 대한 간결한 요약을 나타내며, 아래 규칙을 반드시 따릅니다.

명령형(동사 원형)으로 작성합니다. (예: AddO, AddedX, 추가함X)

첫 글자는 대문자로 작성합니다.

50자 이내로 간결하게 작성합니다.
문장 끝에 마침표(.)를 붙이지 않습니다.

4.2.5. 커밋메시지 예시

- 1) feat (새 기능 추가)
 - feat(api): Implement image-to-video generation flow
 - feat(ui): Add loading spinner during image generation
- 2) fix (버그 수정)
 - fix(download): Stabilize file download using Blob
 - fix(ui): Correct modal rendering error with z-index
- 3) refactor (코드 리팩토링)
 - refactor(prompt): Redesign image prompts by style
 - refactor(api): Separate prompt generation logic into function
- 4) style (코드 스타일 변경)
 - style(ui): Apply uniform shadow to input cards
 - style: Format entire project with Prettier
- 5) chore (기타 작업)
 - chore: Add fal_client to requirements.txt
 - chore: Update .gitignore to untrack .env file

4.3 디자인 시스템 및 UI 가이드

4.3.1. 디자인 원칙

- 서비스의 핵심 가치를 시각적으로 표현하고, 사용자에게 즐거운 경험을 제공하기 위한 4가지 원칙입니다.
- 1) 직관적인 (Intuitive): 모든 기능은 별도의 설명서 없이 탐색하고 사용할 수 있어야 합니다. 사용자의 학습 부담을 최소화하고, 행동의 흐름이 자연스럽게 이어지도록 설계합니다.
- 2) 창의적인 (Creative): 정적인 화면을 넘어, "AI가 기억을 소환한다"는 서비스의 핵심 가치를 시각적으로 표현합니다. 감성적인 문구와 미세한 애니메이션을 통해 사용자의 창의적인 영감을 자극합니다.
- 3) 몰입감 있는 (Immersive): 사용자가 생성한 콘텐츠가 주인공이 되도록 합니다. 어두운 배경의 다크 모드를 기본으로 채택하여, 사용자가 자신의 창작물에 온전히 집중할 수 있는 환경을 제공합니다.
- 4) 피드백 중심 (Feedback-Oriented): 사용자의 모든 행동에 명확한 시각적 피드백을 제공합니다. 로딩 상태, 유효성 검사 실패, 작업 성공 등 각 상황에 맞는 안내를 통해 사용자가 서비스 상태를 즉각적으로 인지할 수 있도록 돕습니다.

4.3.2. 색상 팔레트

역할	색상 코드 (HEX)	색상	설명
포인트 (핑크)	#FF4D8B		'오늘 하루 소환' 버튼, 활성화된 입력창 테두리 등 가장 중요한 행동을 유도하는 요소에 사용합니다.
포인트 (하늘)	#A8CFFF		메인 타이틀의 그라디언트에 사용되어 신비로운 느낌을 더합니다.
배경	#0F1028		배경 이미지 위에 어두운 네이비 색상을 반투명 오버레이로 사용하여 서비스 전체의 몰입감을 조성합니다.
카드 배경	#1E1E1E		글래스모피즘(Glassmorphism) 효과(backdrop-blur)를 적용하여 배경과 콘텐츠를 시각적으로 분리하고 세련된 느낌을 줍니다.
기본 텍스트	#FFFFFF		어두운 배경 위에서 높은 가독성을 보장하기 위해 모든 주요 텍스트에 사용됩니다.
보조 텍스트	#D1D5DB		부가 설명, 비활성화 상태의 텍스트 등 중요도가 낮은 정보에 사용됩니다.
에러	#F87171		파일 업로드 실패 등 유효성 검사에 실패했을 때 사용자에게 경고를 알립니다.
성공	#4ADE80		파일이 성공적으로 선택되었음을 알려주는 등 긍정적인 피드백에 사용됩니다.

4.3.3. 타이포그래피

역할	크기 (데스크탑 / 모바일)	굵기 (Weight)	주요 사용처
페이지 제목	96px / 32px	Bold	MainTitle 컴포넌트의 메인 헤드라인
카드 제목	18px	Semi-bold	GenerationCard 등 각 기능 카드의 제목
본문	16px	Regular	DailyJournalInput의 입력 텍스트
부가 설명	14px	Regular	카드 하단의 안내 문구, 알림 모달의 설명

4.3.4. 레이아웃

- 1) 그리드 시스템
데스크탑 환경에서는 주요 기능 카드들을 3단 그리드로 배치하여 정보를 명확하게 구분하고, 사용자가 쉽게 흐름을 따라갈 수 있도록 돕습니다.
- 2) 컴포넌트 사이의 간격은 40px(gap-10), 카드 내부 여백은 24px(p-6)을 사용하여 시각적으로 안정감 있고 정돈된 느낌을 줍니다.

4.3.5. 핵심 컴포넌트

- 1) 텍스트 입력: 사용자가 하루의 단상을 입력하는 텍스트 영역입니다. 100자로 입력이 제한되며, 텍스트 길이에 맞춰 높이가 자동으로 조절되어 사용 편의성을 높였습니다.
- 2) 옵션 선택: react-select를 커스텀한 드롭다운 메뉴로, 사용자가 원하는 결과물의 화풍, 나이, 젠더를 선택할 수 있습니다. 미 선택 시 랜덤으로 생성됩니다.
- 3) 파일 업로드: 사용자의 얼굴 사진을 선택적으로 업로드하는 영역입니다. 5MB 이하의 이미지 파일만 허용되며, 업로드 시 즉시 미리보기를 제공하여 사용자가 선택한 파일을 바로 확인할 수 있습니다.
- 4) 생성 버튼: 서비스의 핵심 행동을 유도하는 버튼입니다. 로딩 시 텍스트와 아이콘이 "생성 중..."으로 바뀌며, 사용자가 현재 시스템이 동작 중임을 명확히 알 수 있도록 합니다.
- 5) 결과 카드: 생성된 이미지/비디오 썸네일을 보여주는 컴포넌트입니다. 좌우 화살표와 하단의 점(dot) 인디케이터를 통해 생성된 결과물들을 편리하게 탐색할 수 있습니다.
- 6) 알림 모달: 사용자에게 중요한 정보를 전달해야 할 때 나타나는 팝업창입니다. 작업 실패 원인과 해결 방법을 안내하여 막힘없는 서비스 이용을 돕습니다.

4.4 API 컨벤션

4.4.1. 기본 URL (Base URL)

- Production: <https://falsecam.onrender.com>

4.4.2. 일반 규칙

- 1) HTTP 메소드: 모든 API 요청은 POST 메소드를 사용합니다.
- 2) 데이터 형식:
- 3) 요청(Request): 파일이 포함될 경우 multipart/form-data, 그 외에는 application/json 형식을 사용합니다.
- 4) 응답(Response): 모든 응답은 application/json 형식으로 통일합니다.
- 5) 인증 (Authentication):
- 6) 모든 API 요청의 HTTP 헤더에는 sessionId 키로 사용자의 고유 세션 ID(UUID)를 반드시 포함해야 합니다.
- 7) 예시: headers: { 'sessionId': '...' }

4.4.3. 표준 응답 구조 (Standard Response Structure)

API 응답은 항상 아래와 같은 JSON 구조를 따릅니다.

1) 성공 시 (Success)

JSON

```
{
  "success": true,
  "data_key": "..." // 예: "image", "video_url"
}
```

2) 실패 시 (Error)

JSON

```
{
  "success": false,
  "error": "에러 원인에 대한 설명 메시지"
}
```

4.4.4. API 명세 (Endpoints)

1) 이미지 생성 API

- 기능: 사용자의 입력(텍스트, 옵션, 이미지 파일)을 받아 AI 이미지를 생성하고 Base64 데이터 URI로 반환합니다.

- Method: POST

- Endpoint: /generate/image

- Headers:

sessionID: string (필수)

- Request Body (multipart/form-data):

text: string (사용자 일기)

style: string (e.g., "realistic", "2d")

age: string (e.g., "9", "20")

gender: string (e.g., "male", "female")

image: File (이미지 파일, 선택 사항)

- Success Response (200 OK):

JSON

```
{
  "success": true,
  "image": "data:image/png;base64,..."
}
```

- Error Response (500 Internal Server Error):

JSON

```
{
  "success": false,
```

```
"error": "API key missing."
}
```

2) 비디오 생성 API

- 기능: 생성된 이미지(Base64 데이터 URI)와 프롬프트를 받아 AI 비디오를 생성하고 비디오 파일의 URL을 반환합니다.

- Method: POST

- Endpoint: /generate/video

- Headers:

Content-Type: application/json

sessionID: string (필수)

- Request Body (application/json):

JSON

```
{
  "prompt": "string",
  "image_url": "string (Base64 Data URI)",
  "style": "string",
  "age": "string",
  "gender": "string"
}
```

- Success Response (200 OK):

JSON

```
{
  "success": true,
  "video_url": "https://.../video.mp4"
}
```

- Error Response (500 Internal Server Error):







JSON

```
{
  "success": false,
  "error": "Video generation failed."
}
```

제5장 배포

- 프로젝트는 GitHub를 중심으로 프론트엔드와 백엔드가 각각 Cloudflare Pages와 Render를 통해 자동 배포됩니다. CI/CD 파이프라인을 통해 코드 변경 사항이 main 브랜치에 푸시될 때마다 즉시 서비스에 반영되는 효율적인 개발 및 배포 환경을 구축했습니다.

5.1 Cloudflare Pages 설정

화면명	Cloudflare 빌드 구성	
환경 선택:	프로덕션 ▼	
빌드		
Git 리포지토리	 bravery33/FalseCam	관리 
빌드 구성	빌드 명령: npm run build 빌드 출력: dist 루트 디렉터리: 빌드 메모: 사용됨	
빌드 캐시 베타	사용 안 함	사용
분기 제어	프로덕션 분기: main 자동 배포: 사용됨	
빌드 감시 경로	경로 포함: *	
빌드 시스템 버전	버전 3	
배포 후크	정의된 배포 후크 없음	+

변수 및 암호

프로젝트의 텍스트, 비밀 또는 빌드 변수를 정의

+ 추가

편집

유형	이름	값		
비밀	BFL_API_KEY	암호화된 값	✎	🗑
비밀	GOOGLE_APPLICATION_CREDENTIALS_JSON	암호화된 값	✎	🗑
비밀	HF_API_KEY	암호화된 값	✎	🗑
비밀	OPENAI_API_KEY	암호화된 값	✎	🗑

빌드 구성

리포지토리 관련 빌드 설정을 구성합니다.

[설명서](#)

프레임워크 미리 설정



React (Vite)



빌드 명령

npm run build

빌드 출력 디렉터리

/

dist

> 루트 디렉터리(고급)



빌드 메모 사용

공개 풀 요청 시 Cloudflare가 빌드 상태 및 미리 보기 링크와 함께 빌드 메모를 추가합니다.

설명

- 프레임 워크 설정: React (Vite)
- 빌드 명령: npm run build
- Cloudflare 배포 완료 링크: <https://falsecam.pages.dev/>

5.2 Render 설정

화면명	Render 설정
<h3>Settings</h3> <div><h4>General</h4><div><div>Name A unique name for your Web Service.</div><div>FalseCam</div><div>Edit</div></div><div><div>Region Your services in the same region can communicate over a private network.</div><div>Singapore (Southeast Asia)</div></div><div><div>Instance Type</div><div><div>Free 0.1 CPU 512 MB</div><div>Update</div></div><div>Please ask the owner (diddung83@gmail.com) to enter their payment information.</div><div>See remaining free usage, or learn about free service limits.</div></div></div>	
<h3>Build & Deploy</h3> <div><div>Repository The repository used for your Web Service.</div><div>https://github.com/bravery33/FalseCam</div><div>Edit</div></div> <div><div>Branch The Git branch to build and deploy.</div><div>main</div><div>Edit</div></div> <div><div>Git Credentials User providing the credentials to pull the repository.</div><div>diddung83@gmail.com (you)</div><div>Use My Credentials</div></div> <div><div>Root Directory Optional If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo.</div><div>BackEnd/</div><div>Edit</div></div>	

Build Filters

Include or ignore specific paths in your repo when determining whether to trigger an auto-deploy. Paths are relative to your repo's root directory. [Learn more.](#)

Included Paths

Changes that match these paths will trigger a new build.

+ Add Included Path

Ignored Paths

Changes that match these paths will not trigger a new build.

+ Add Ignored Path

Build Command

Render runs this command to build your app before each deploy.

Backend/ \$ pip install -r api/requirements.txt

Edit

Pre-Deploy Command Optional

Render runs this command before the start command. Useful for database migrations and static asset uploads.

Backend/ \$

Edit

Start Command

Render runs this command to start your app with each deploy.

Backend/ \$ uvicorn api.server:app --host 0.0.0.0 --port \$PORT

Edit

Auto-Deploy

By default, Render automatically deploys your service whenever you update its code or configuration. Disable to handle deploys manually. [Learn more.](#)

On Commit

Edit

Deploy Hook

Your private URL to trigger a deploy for this server. Remember to keep this a secret.

Regenerate hook

설명

- Build Command: pip install -r api/requirements.txt
- Start Command: uvicorn api.sever:app --host 0.0.0.0 --port \$PORT
- Render 배포 완료 링크: <https://falsecam.onrender.com>

제6장 결 론

6.1 프로젝트 성과 및 의의

‘FalseCam’ 프로젝트는 단순히 최신 AI 기술을 활용해보는 것을 넘어, 하나의 아이디어를 구체적인 기획(기획서, WBS)으로 발전시키고, 실제 동작하는 서비스로 구현하여 배포한 뒤, 실사용자 피드백을 통해 개선하는 소프트웨어 개발의 전 과정(Full-cycle)을 성공적으로 완주했다는 데 가장 큰 의의가 있습니다. 체계적인 WBS 기반의 일정 관리와 꾸준한 회의(회의록)는 한정된 시간 속에서도 팀의 역량을 집중하고 프로젝트를 목표 일정 내에 마무리할 수 있었던 원동력이었습니다.

본 프로젝트는 Midjourney, Sora 등 기존의 전문적인 AI 이미지/영상 생성 서비스와는 명확히 다른 목표를 가집니다. 복잡한 영어 프롬프트나 전문 지식을 요구하는 서비스들과 달리, FalseCam은 별도의 학습 없이 누구나 자신의 일상 기록만으로 쉽고 재미있게 창작물을 만들 수 있도록 진입 장벽을 낮추는 데 집중했습니다. 이 ‘경험 재구성’이라는 차별화된 접근은 기술 구현의 방향성을 명확히 해주었습니다.

기술적으로는 프로젝트의 핵심인 생성형 AI 모델을 단순히 사용하는 데 그치지 않고, 여러 모델(Flux-Kontext-Pro, Kling 등)을 직접 테스트하며 품질, 속도, 비용을 종합적으로 분석하여 최적의 모델을 선정하는 실증적인 연구를 수행했습니다. 개발 과정에서는 실제 웹 서비스 환경에서 마주할 수 있는 기술적 난제들을 직접 해결하며 역량을 키웠습니다. 브라우저의 보안 정책으로 인해 발생한 CORS 문제를 Blob 객체를 활용해 파일 다운로드 기능을 구현하여 해결했으며, 외부 AI API 연동 시 발생한 base64 인코딩 문제와 z-index로 인한 UI 렌더링 오류 등 구체적인 문제들을 해결하며 탄탄한 기본기를 다졌습니다. 또한, GitHub Actions와 연동된 CI/CD 파이프라인을 구축하여 코드 변경 사항이 Cloudflare Pages와 Render에 자동으로 배포되는 효율적인 개발 문화를 구축한 것 역시 큰 성과입니다.

무엇보다 이 프로젝트의 가장 빛나는 성과는 실사용자 피드백을 중심으로 서비스를 발전시킨 경험입니다. MVP 배포 후 총 14명의 사용자로부터 피드백을 수집했으며, "가볍게 재미로 하기 좋은 앱 같다", "단순하고 직관적이라 좋다" 등의 긍정적인 반응을 통해 프로젝트의 핵심 가치가 사용자에게 잘 전달되었음을 확인했습니다. 동시에 "예시 멘트가 추상적이다", "어디에 기입해야 할지 한눈에 파악이 어렵다" 등 날카로운 지적들 또한 귀담아들었습니다. 우리 팀은 이 피드백을 즉각 반영하여, 사용자가 입력할 내용의 예시를 보여주는 플레이스홀더(Placeholder) 문구를 더욱 구체적이고 명확하게 수정하는 등 실질적인 UI/UX 개선을 진행했습니다. 이처럼 사용자와의 소통을 통해 서비스를 더 나은 방향으로 발전시킨 경험은 이번 프로젝트의 가장 큰 보람이자 소중한 자산입니다.

- MVP 사용자 피드백 결과 요약



6.2 향후 발전 방향

이번 MVP는 FalseCam의 핵심 가치를 증명하는 시작점이며, 사용자 피드백과 팀의 비전을 바탕으로 다음과 같은 방향으로 서비스를 고도화할 계획입니다.

6.2.1. 사용자 경험의 개인화: 회원 시스템 및 마이페이지 도입

- 현재의 세션 기반 서비스는 사용자가 생성한 소중한 결과물을 영구적으로 보관할 수 없다는 한계가 있습니다. 이를 해결하기 위해 정식 회원 시스템과 마이페이지를 도입하고자 합니다. 사용자는 자신만의 계정을 통해 과거에 생성했던 모든 이미지와 영상을 '추억 아카이브'처럼 모아볼 수 있게 됩니다. 마이페이지에서는 생성물을 날짜별로 정렬하거나, 다시 다운로드하고, 원하는 결과물만 삭제하는 등 체계적인 관리가 가능해져, 일회성 체험을 넘어 사용자와 서비스 간의 지속적인 관계를 형성할 것입니다.

6.2.2. 창작의 자유도 확장: 스타일 추가

- 사용자가 더 다채로운 결과물을 만들 수 있도록 선택할 수 있는 스타일을 지속적으로 추가할 계획입니다. 현재 제공되는 '실사', '2D 애니메이션' 등의 기본 스타일 외에도, '수채화', '스케치'와 같은 새로운 화풍을 더하여 사용자가 자신의 하루를 더욱 다양한 모습으로 재창조할 수 있도록 선택의 폭을 넓히고자 합니다.

6.2.3. 콘텐츠 경험 고도화

1) 그림 일기 형식 지원

사용자의 기록을 한 편의 완성된 이야기처럼 느낄 수 있도록 ‘일기 형식’을 정식으로 지원할 계획입니다. 이 기능은 상단에는 AI가 생성한 그림이, 하단에는 AI가 재구성한 문장이 함께 표시되는 새로운 레이아웃을 제공합니다. 사용자는 원할 경우, 이 정적인 그림을 감성적인 움직임이 더해진 짧은 동영상으로 변환하여 그날의 기억을 더욱 생생하고 특별하게 간직할 수 있습니다.

2) 텍스트 재구성 기능 도입

FalseCam의 핵심 가치인 ‘경험 재구성’을 극대화하기 위해, ‘텍스트 왜곡(재구성)’ 기능을 도입하고자 합니다. 이 기능은 사용자의 평범한 일상을 전혀 다른 장르의 특별한 사건으로 재창조하는 것입니다. 예를 들어, "카페에서 커피를 마셨다"와 같은 단순한 기록을 입력하면, AI가 이를 "고대 마법사들의 비밀 서재, 나는 룬 문자가 새겨진 잔에 담긴 검은 지혜의 물약을 마셨다"와 같이 환상적인 판타지 속 한 장면으로 완전히 왜곡하고 확장해주는 식입니다. 이를 통해 사용자는 단순한 기록 행위를 넘어, AI와 함께 자신의 하루를 영화나 소설처럼 재창조하는 새로운 창작의 즐거움을 경험하게 될 것입니다.

6.3 맺음말

‘FalseCam’ 프로젝트는 단순한 포트폴리오를 넘어, 아이디어가 실제 서비스로 구현되고 사용자에게 전달되는 전 과정을 직접 경험한 값진 여정이었습니다.

“평범한 하루를 특별한 추억으로 만들 수 있을까?”라는 작은 호기심에서 출발했지만, 기획과 설계, 수많은 기술적 난관, 그리고 사용자 피드백까지 모든 단계가 결코 쉽지 않았습니디다. 처음엔 각자 다른 배경과 스타일 때문에 팀워크가 쉽지 않았지만, 시간이 지날수록 서로를 이해하며 협업의 힘을 실감할 수 있었습니다.

UI 오류(z-index), 복잡한 비동기 API 연동(CORS, Blob) 등 예상치 못한 문제로 밤늦게까지 해결책을 고민했고, 사용자의 작은 불편도 놓치지 않으려 프롬프트와 UI를 계속 다듬었습니다. 그 과정에서 의견 충돌도 있었지만, 토론을 통해 더 나은 선택을 할 수 있었습니다.

무엇보다, ‘가볍고 재미있는 AI 경험’이라는 사용자들의 긍정적인 반응이 이번 프로젝트의 가장 큰 보람이었습니다. 이는 기술 구현에만 집중하지 않고, 사용자 입장에서 꾸준히 고민하며 서비스를 개선해 나간 결과라고 생각합니다.

이번 프로젝트를 통해 기술적인 자신감과 협업의 소중함, 그리고 사용자를 향한 진심이 무엇인지 배웠습니다. FalseCam은 끝났지만, 앞으로도 누군가의 일상에 작은 즐거움과 가치를 더할 수 있는 개발자로 계속 성장해 나가고 싶습니다.