한신대학교

AI · SW캡스톤디자인 계획서 (제안서)

프로젝트	**제 목**	AI 기반 허위정보 수집·검증 및 팩트체크 플랫폼		
팀장	**팀 명**	FAITH (페이스): Fact + AI + Truth + Humanity		
	성 명	왕희원	**학번**	202378040
	연락처	010-5148-7456		
	E-MAIL	gmldnjs0321@hs.ac.kr		
구분	**성 명**	**학번**	**E-MAIL**	**연락처(H.P)**
팀원 인적사항	진수빈	202378073	vheh5053@hs.ac.kr	010-8237-3792
	김나영	202378176	sinamon500@hs.ac.kr	010-3050-6266
	심서윤	202378188	cmsewyoon@hs.ac.kr	010-3290-6326
지도교수	홍승필			

본인과 팀원은 2025학년도 2학기 AI·SW캡스톤디자인1 과목의
프로젝트에 대한 캡스톤디자인 계획서를 다음과 같이 제출합니다.

2025 년 12월 4일

팀 장 : 왕희원 (서명)

한신대학교 AI · SW대학

목 차

1. 문제 및 목적

현대 사회는 디지털 미디어의 발전으로 정보의 생산과 확산 속도가 급격히 증가함에 따라 기술을 악용한 허위 정보의 확산이 심각한 사회 문제로 떠오르고 있다. 특히 인공지능 기반의 딥페이크 기술은 실제와 구분하기 어려운 조작 이미지와 영상을 손쉽게 생성함으로써, 개인의 명예를 훼손하는 등 사회적 혼란을 가져오는 **범죄**로 작용하고 있다.

영국 BBC는 2024년 ChatGPT, Copilot, Gemini, Perplexity 등 4종의 AI 챗봇을 대상으로 100건의 기사 요약 및 질의응답 테스트를 진행한 결과, 51%에서 오류·왜곡·허위 정보가 포함됐다고 보도했으며, 유튜브나 인스타그램 등의 SNS에서도 생성형 AI를 이용한 합성 이미지, 영상 등이 AI 표기 없이 유포되고 있어 사용자에게 하여금 혼돈을 유발하고 있다.

이러한 허위 정보는 개인의 판단력을 흐리게 하고 사회 전반의 신뢰를 약화하며, 공공 안전과 민주적 의사결정에도 부정적인 영향을 미친다.

기존의 언론사나 전문 기관 중심의 팩트체크 시스템은 정보의 확산 속도를 따라가지 못하고, 일반 사용자의 참여가 제한적이라는 한계를 지닌다. 또한, 대부분의 검증 과정이 사후적으로 이루어져 실시간 대응이 어렵고 딥페이크나 이미지 조작과 같은 비정형 데이터에 대한 분석이 충분히 이루어지지 못하고 있다.

특히, 국내에서는 정치 분야 중심의 해외 기관 의존형 팩트체크 서비스가 주를 이루고 있어, 경제, 사회, 문화 등 다양한 분야를 포괄하는 한국 특화형 검증 서비스가 부족한 상황이다. 또한, 사용자 참여를 유도할 수 있는 보상 구조나 접근성 개선이 미비하여 대중적 활용도가 낮고, 서비스의 신뢰성 또한 충분히 확보되지 못하고 있다.

이러한 한계는 단순한 기술적 대응만으로는 극복하기 어려우며, AI 기술과 시민 참여를 결합한 새로운 검증 체계의 구축이 필요함을 보여준다.

본 프로젝트는 '**AI 기반 허위 정보 수집·검증 및 팩트체크 플랫폼**'을 구축하여 인공지능 기술을 활용한 자동 검증과 사용자 참여형 구조를 통해 신뢰할 수 있는 정보 환경을 조성하는 것을 **목표**로 한다. 또한, 허위 정보로 인한 사회적 피해를 줄이고 건강한 디지털 소통 문화를 형성함으로써 지속 가능한 정보 생태계 구축에 이바지할 것이다.

2. 관련 연구

1) 시장 분석 : 시장 상황과 전망

① 생성형 AI 관련 피해 및 사회적 문제와 법적 분쟁

최근 생성형 AI(예: 딥페이크, 가짜 합성 뉴스 등)의 확산으로 인한 실제 피

해 사례가 급격히 증가하고 있다. 대표적으로 딥페이크 성범죄는 피해자가 영상 존재조차 모른 채, 합성된 영상이 온라인에 유포되어 심각한 2차 피해를 겪는 경우가 많다.

또한 AI 생성 허위 정보도 큰 문제로 대두되고 있다. EU 선거를 앞두고 AI가 생성한 합성 콘텐츠나 가짜 사이트가 선전 목적으로 활용되며, 미국 뉴스 앱 NewsBreak에서는 AI가 사실과 다른 내용을 기사로 생산하는 사례가 발생했다. 이처럼 AI는 스스로 생성한 정보의 진위 여부를 판별하지 못하기 때문에 콘텐츠 신뢰성의 근본적 위기가 제기되고 있다.

AI 생성물로 인한 저작권 및 초상권 분쟁도 증가하고 있다. 디즈니는 AI 플랫폼 'Midjourney'를 저작권 침해 혐의로 고소하였고 Getty Images 역시 Stability AI를 상대로 법적 조치를 취하였다.

음악 산업에서도 AI 생성물에 대한 저작권 침해 소송이 다수 제기되는 등, AI 창작물의 법적 경계가 불분명한 상황이다.

② 기술 및 정책 동향

- 딥페이크 탐지 기술: 이미지·영상 합성 여부를 판별하는 AI 기반 기술이 다양하게 개발되고 있으며, Reality Defender 등의 서비스가 상용화되어 있다.
- 워터마킹 및 출처 증명: OpenAI와 Adobe 등이 주도하는 C2PA 표준이 확산 중이며, AI 콘텐츠의 신뢰성과 투명성을 확보하기 위한 기술적 기반이 마련되고 있다.
- 플랫폼 정책 변화: 유튜브는 '변경되거나 합성된 콘텐츠'를 업로드할 때 반드시 이를 공개하도록 정책을 개정하였다.

- AI 생성 텍스트 탐지: Originality.ai 등 다양한 탐지기가 등장했으나, 완전 자동화보다는 다중 탐지 + 사람 검수형 접근이 주류를 이루고 있다.

③ 시장 공백 및 차별화 방향

- 한국어 특화 서비스의 부재

현재 대부분의 AI 탐지·팩트체크 서비스는 영어 기반으로 운영되며, 한국어 콘텐츠에 특화된 플랫폼은 거의 없다. 따라서 한국어 모델 최적화가 본

프로젝트의 주요 경쟁력이 될 수 있다.

- 법·정책 연동형 라벨링 필요

생성형 AI 콘텐츠에 법적 고지나 위험 안내 라벨을 표시함으로써 사용자에게 AI 오용에 대한 경각심을 제공할 수 있다.

- 접근성 개선 방안

웹사이트 형태의 검증 서비스는 접근성이 낮기 때문에 카카오톡 채널 연동을 통한 손쉬운 제보·검증 접근을 고려할 수 있다.

- 출처 증명 및 API 연동 확대

C2PA, Reality Defender 등 외부 API를 통합하여 다중 검증 체계-출처 증명과 AI 탐지, 검수자 확인 파이프라인을 구축하면 기존 서비스와 차별화된 신뢰 기반 시스템을 형성할 수 있다.

④ 결론

현재 시장은 생성형 AI의 빠른 발전 속도를 따라가지 못하고 있으며, 법적 규제와 기술 표준이 미비한 상태이다. 이에 따라 한국어 기반 AI 탐지 및 검증 플랫폼의 시장 진입 여지는 충분하며, 기존 해외 서비스들이 제공하지 못하는 언어·정책·접근성 측면의 현지화 전략이 본 프로젝트의 핵심 경쟁력으로 작용할 수 있다.

2) 유사 시스템

① 해외 사례

(1) Sensity AI

(가) sensity

Sensity AI는 이미지, 영상, 오디오 등 다양한 미디어를 분석하여 딥페이크 여부를 판별하는 AI 기반 멀티모달 탐지 플랫폼이다. 사용자가 콘텐츠를 업로드하면 AI가 수초 내에 조작 가능성을 평가하며, 클라우드·온프레미스 환경을 모두 지원한다. 또한, 외부 서비스와의 연동을 위한 API 통합 기능을 제공하여, 기업이나 기관이 자체 시스템에 탐지 기능을 손쉽게 적용할 수 있다. 현재 유럽을 중심으로 미디어 기업, 보안 기관 등에서 활용되고 있으며, 딥페이크 대응의 산업 표준화 방향을 제시하고 있다.

(2) Google Fact Check Tools

(가) Google Fact Check Tools

Google Fact Check Tools는 전 세계 팩트체크 기관이 게시한 기사에 포함된 ClaimReview 마크업을 통합하여 검색·조회할 수 있는 도구 세트다. 일반 사용자는 Fact Check Explorer를 통해 주제나 키워드별로 검증된 팩트체크 기사를 탐색할 수 있으며, 개발자는 Claim Search API를 이용해 동일한 데이터셋을 프로그램적으로 질의할 수 있다. 이 플랫폼은 구조화된 ClaimReview 데이터를 기반으로 허위 정보 확산을 방지하고, 언론인·연구자·기관이 신뢰할 수 있는 사실 검증 정보를 손쉽게 활용할 수 있도록 지원한다.

② 국내 사례

(1) SNU 팩트체크

(가) SNU factcheck 서울대학교 언론정보연구소

SNU 팩트체크는 언론사와 서울대학교 언론정보연구소가 협력하여 출범시킨 협업형 팩트체킹 모델로, 언론사들이 팩트체크한 내용을 게시할 수 있도록 플랫폼을 제공한다.

본 플랫폼은 정보의 '사실'에 대한 기본적인 관점을 설명하고, 메타데이터 분석을 통해 '팩트'의 형식적 구성 요건에 관한 탐색적 연구를 진행하기도 했다. 또한, SNU 팩트체크 정보가 특정 수치 기반의 종속 변수 측정에 이용되는 등 다양한 학술 연구에서 주요 데이터로 활용된 바 있다.

SNU 팩트체크는 인공지능 기반의 가짜 뉴스 시스템 연구에 뉴스 데이터를 활용되며, 국내 소셜 미디어 환경에서 트윗 데이터를 기반으로 트윗 내용의 거짓 여부를 판별하는 자동화된 팩트체크 연구에서도 활용되었다. 다만, 2024년 재정난으로 인해 운영이 무기한 중단되면서, 공익 기반 팩트체크 플랫폼의 지속가능성과 재원 구조의 한계가 드러났다는 평가가 제기됐다.

(2) 팩트체크넷(FactCheckNet)

(가) 팩트체크넷

팩트체크넷(FactCheckNet)은 한국언론진흥재단이 2020년에 출범시킨 공공

형 팩트체크 플랫폼으로, 국내 언론사들이 수행한 팩트체크 결과를 통합해 제공하는 아카이브형 서비스이다.

각 언론사에서 개별적으로 운영되던 팩트체크 기사를 한곳에서 검색할 수 있도록 지원하며, '사실', '사실이 아님' 등 검증 결과를 표준화된 등급 체계를 도입했다. 또한 관련 기사 링크 제공 등 팩트체크를 보조하는 다양한 기능을 제공하며 팩트체크 과정 및 결과를 투명하게 공개했다.

팩트체크넷의 중요한 목표 중 하나는 시민과 소통하며 그들의 요구를 충족시키는 플랫폼이 되는 것이다. 시민들은 전문가, 즉 기자에게 질문을 던지고, 팩트체크 과정에 참여하여 사실 검증의 주체로서 역할을 할 수 있다.

구조는 언론사 중심으로 구성되어 일반 사용자의 직접적인 참여가 제한적이며, 정부 재원에 의존하는 운영 형태로 인해 지속 가능성 확보가 어렵다. 또한, 2023년 2월 28일에 정부 예산 축소와 언론사 참여 저조로 인해 서비스가 종료된 상태이다.

3) 기존 탐지 방식의 기술적 한계 및 본 연구의 차별성

기존의 가짜 정보 탐지는 규칙 기반과 인력 중심 검증에 의존해 정확도와 확장성에 한계가 있었다. 반면, 에이전트 기반 AI 탐지 방식은 텍스트·이미지·외부 근거를 자동 결합하여 더 빠르고 높은 정확도의 실시간 검증을 제공할 수 있을 것이다.

구분	기존의 탐지 방식	에이전트 기반 AI 탐지 방식
탐지 방식	규칙 중심, 전문가 검토 기반	다중에이전트가 텍스트·이미지·외부 근거를 자동 분석
처리 속도	분석 지연, 수작업 중심	실시간 또는 준실시간 탐지 가능
확장성	인력·자원 증가 없이 확장 어려움	자동화 구조로 대량 콘텐츠 처리 용이
정확도	단일 정보(텍스트)에 치우쳐 오류 잦음	멀티 모달 기반 분석으로 고정확 탐지 수행
비용 효율성	검증 비용 높고 반복 작업 필요	자동화 기반 운영 비용 절감
신규 위협 대응력	새로운 조작·패턴 반영에 느림	모델·에이전트가 신속하게 패턴 업데이트
기술 연동성	기존 시스템과 통합이 제한적	API·클라우드와 쉽게 연동 가능

4) 시스템 구축 관련 기술

① Front-end

(1) Next.js vs. React

Next.js란 Vercel 사에서 개발한 React로 만드는 SSR FrameWork로, 풀스택 웹 애플리케이션을 지원한다.

React와 Next.js의 가장 큰 차이점은 CSR(React)과 SSR(Next.js)이다. CSR(Client Side Rendering)과 SSR(Server Side Rendering)은 사용자가 브라우저에서 보는 화면인 UI를 어디서 만들어 주는지에 따라 구분된다. CSR은 Clinet(Front-end), SSR은 Server(Back-end)에서 화면을 구성한다.

React는 브라우저가 JS 코드를 가지고 있지 않거나, 요청 중인 상태라면 UI를 구성할 수 없고, 사용자는 빈 화면을 보게 된다. React 코드가 실행되기 전까지 사용자 화면에는 아무것도 보이지 않는 것으로, 클라이언트 측에서 UI를 제작하는 것을 CSR 방식이라 한다. React를 사용할 경우 초기 로드만 완료되면 이후 렌더링이 빠르며, 서버에 요청할 것이 거의 없어 서버 부담이 적다. 하지만, 초기 로드가 오래 걸리고 외부 라이브러리가 필요한 경우가 많으며 SEO에 좋지 않다.

Next.js는 서버에서 UI를 모두 구성한 후 사용자에게 응답해 화면을 보여주는 방식으로, 화면이 pre-rendering되어 사용자는 인터넷 속도에 상관없이 화면에 뭔가 나오는 것을 볼 수 있다. 이렇게 서버 측에서 UI를 렌더링하는 것을 SSR 방식이라 한다. Next.js를 사용할 경우 초기 로딩 속도가 빠르며 SEO에 좋다. 하지만, 서버에서 전체 앱을 미리 rendering 하므로 컴포넌트 로딩이 오래 걸리면 사용자는 빈 화면을 볼 수 있다. 게다가 페이지를 요청할 때마다 새로고침 되어 UX가 다소 떨어지며 서버에 매번 요청하기 때문에 서버 부하가 크다.

② Back-end

(1) Web FrameWork

(가) FastAPI

FastAPI는 Python 기반의 최신 API 프레임워크로 타입 힌팅과 Pydantic을 이용해 입력을 안전하게 검증하고 OpenAPI 문서를 자동 생성해 주는 점이 핵심이다. 내부적으로 ASGI 비동기 처리를 지원해서 외부 API 호출이나 I/O가 많은 AI 백엔드에서 성능 이점을 주고, 타입 기반 설계 덕분에 코드 읽기·디버깅·자동완성 경험이 좋아 팀 개발 속도를 올려준다. 자동문서와 타입 안정성 때문에 빠르게 안정적인 API를 만들기 좋다.

다만 관리자 UI나 ORM 같은 배터리 포함 기능은 내장되어 있지 않으니

필요하면 SQLAlchemy나 별도 대시보드 라이브러리를 추가해야 하고, 팀에 생소하면 초기 설계 규칙을 잘 정해 두는 편이 안전하다.

(나) Flask

Flask는 아주 가벼운 WSGI 프레임워크로 학습 곡선이 낮고 프로토타입·PoC를 빨리 만들 때 적합하다. 플러그인 에코시스템이 풍부해서 필요한 기능을 골라 붙일 수 있고, 단순한 서비스 구조로 팀이나 개인이 빠르게 결과물을 내기 좋다.

주의할 점은 기본이 동기 모델이라 비동기 워크로드가 많아지면 Celery 같은 작업 큐나 ASGI 별도 구성이 필요하다는 점이고, 서비스가 커질 경우 코드 구조와 규약을 강제하지 않으면 유지보수가 힘들어질 수 있다는 것이다.

(2) DataBase

(가) PostgreSQL

PostgreSQL은 오픈소스 객체-관계형 DB로 ACID를 보장하고 JSONB·Full-Text Search(FTS) 같은 기능을 제공해 반정형 데이터와 텍스트 검색을 함께 다루기 좋다. 판정 로그·메타데이터 같은 정합성이 중요한 데이터는 PostgreSQL에 저장하고, JSONB를 활용해 원문 메타를 병행 보관하면 구조·유연성 모두 잡을 수 있다. 다만 대규모 읽기 부하가 발생하면 캐시(예: Redis)나 전용 검색 인덱스(Elasticsearch 등)를 도입해 보완해야 한다.

(나) MongoDB

MongoDB는 문서 지향 NoSQL로 스키마 유연성이 높아 다양한 메타·원문을 빠르게 적재하고 빠른 반복 개발을 하기에 편리하다. 초기 개발단계에서 스키마 변경이 잦을 때 개발 생산성을 크게 올려주고 샤딩을 통한 수평 확장도 지원한다.

반면 복잡한 조인이나 엄격한 트랜잭션(금융/회계 등)이 필요하면 제약이 있으니 그런 워크로드에는 적합하지 않을 수 있다.

(3) ORM

(가) SQLAlchemy

SQLAlchemy는 Python에서 널리 쓰이는 ORM으로 선언적 모델과 세션 기반 트랜잭션 관리, 복잡한 쿼리 표현을 지원한다. FastAPI와 궁합이 좋아 Pydantic을 통한 입력 검증과 함께 사용하면 타입 안정성 있는 API를 만들기 쉬우며 DB 엔진 교체 시 유연성이 높다.

(나) Django ORM

Django ORM은 Django에 내장되어 있어 모델 정의·마이그레이션·관리자 연동이 매끄럽고 CRUD 중심 개발에서 높은 생산성을 제공한다. ORM이 SQL 인젝션 같은 기본 위협을 완화해 주는 점도 장점이다.

③ MAS

MAS(Multi-Agent System)은 이름 그대로 다수의 에이전트가 동시에 존재하면서 상호작용하는 시스템을 말한다. 여기서 말하는 'Agent'는 독립적으로 사고하고 행동할 수 있는 존재로, 환경과의 상호작용을 통해 목표를 달성하려는 주체이다. 쉽게 말하면, MAS에서는 여러 Agent가 각자 독립적이지만 동시에 환경을 공유하면서 함께 문제를 해결하거나 경쟁하는 구조로 되어 있다.

MAS에는 Agent, 환경, 상호작용이라는 3가지 축으로 구성된다.

Agent는 MAS의 핵심 주체로, 환경을 관찰 및 상태 해석 후 특정 행동을 선택한다. 이 과정은 보통 정책이라는 함수로 정의된다. 각 Agent는 개별의 목적, 관점, 행동전략을 갖고 있으며, 시스템 안에서 독립적으로 또는 협력적으로 작동한다.

환경은 모든 Agent가 공유하는 시뮬레이션 공간이다. Agent는 관측을 통해 정보를 받아들이고 행동을 통해 환경을 바꾼다. 이렇게 상호작용을 하면서 각자의 목적을 달성하려 한다.

MAS의 핵심은 상호작용이다. Agent는 단순히 환경만 바꾸는 게 아니라, 다른 Agent의 행동에도 영향받는다. 이 상호작용은 협업, 경쟁, 혹은 2개 다일 수도 있으며 보상을 공유하거나, 팀을 이뤄 행동할 때는 더욱 복잡한 전략이 필요하다.

MAS는 단순한 Agent의 묶음이 아니라 상호작용을 통한 복잡한 전략과 행동의 집합체라고 볼 수 있다.

(1) Lagnchain Agent vs. CrewAI

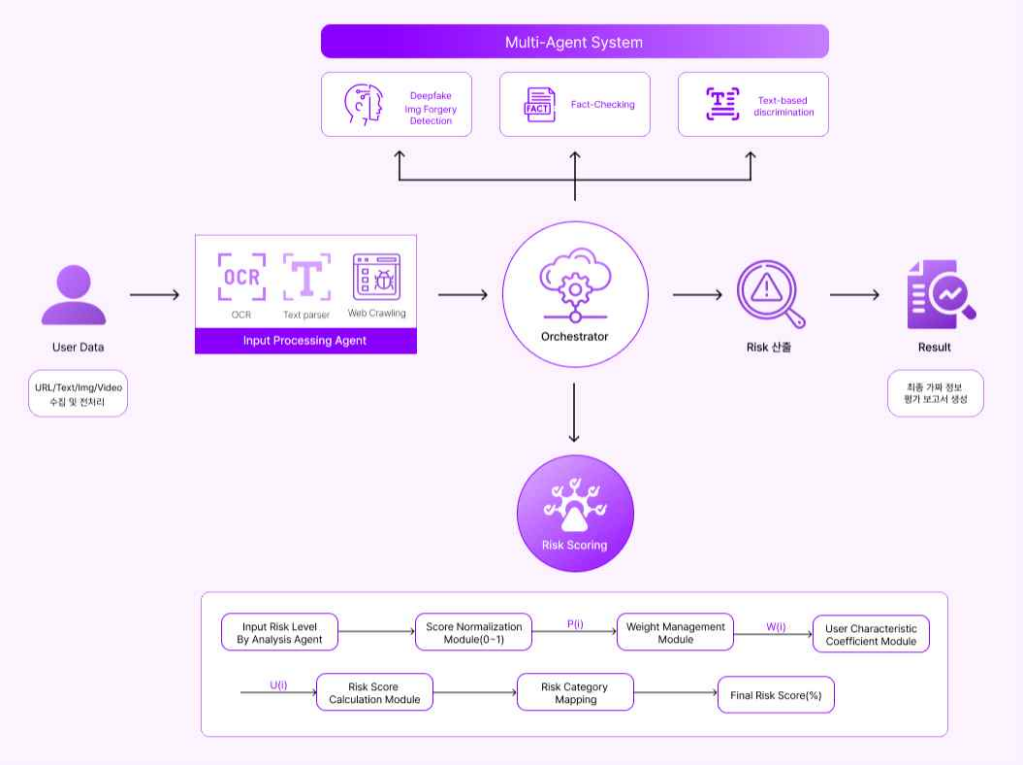
Langchain Agent는 다양한 외부 도구와 데이터 소스, 데이터베이스를 통합하는 생태계가 잘 구축되어 있어, 한국어 RAG 기반 팩트체킹, 검색 및 DB 연동, API 호출 등 "하나의 Agent가 여러 도구를 적절히 선택해 사용하는" 시나리오에 적합하다.

CrewAI Agent는 여러 Agent를 역할과 목표에 따라 나누고, Agent 간 협업을 통해 복잡한 워크플로우를 수행하는 Multi-Agent FrameWork로, 기획/검증/보고서 작성 등 사람 조직과 유사한 구조를 구현하기에 용이하다. 따라서 Langchain은 주로 도구 통합과 RAG 파이프라인 구성에 강점을 가지며, CrewAI는 역할 분리와 협업이 중요한 Multi-Agent 시스템 설계에 강점을 갖는 것으로 평가된다.

본 프로젝트에서는 향후 서비스 확장성과 데이터 통합 측면을 고려하여 Langchain Agent를 기본 프레임워크로 채택한다. Langchain은 벡터 DB, 검색 엔진, 외부 API 등과의 연동이 이미 표준화되어 있어, 한국어 팩트체킹을 위한 RAG 파이프라인을 빠르게 구축할 수 있다. CrewAI는 Multi-Agent WorkFlow 설계에 장점이 있지만, 본 프로젝트의 1차 목표는 '정확한 정보 검증과 데이터 파이프라인 구축'에 있으므로, 우선 Langchain 기반의 단일/소수 Agent 구조를 채택한 뒤, 필요시 Multi-Agent 시나리오를 LangGraph로 확장하는 것을 계획한다.

3. 제안 방법

1) Risk scoring 기반 위험도 평가 엔진 설계



본 연구에서는 텍스트·이미지·영상 패턴 분석 에이전트로부터 전달된 다양한 위험 신호(%)를 하나의 '**최종 위험도**(Risk Score)'로 통합하는 Risk Scoring 엔진을 설계한다. 제안 엔진은 콘텐츠 자체의 위험 신호뿐 아니라, 사용자 특성 정보를 반영한 맥락 기반 위험도를 고려한다. 이를 통해 단순한 분류 결과가 아닌, 사용자에게 실질적으로 의미 있는 정량적 위험 지표를 제공한다.

각 분석 모듈이 산출한 점수(예: 허위 정보 확률, 딥페이크 가능성, 사기 패턴 점수 등)를 정규화한 뒤, 가중합(Weighted Sum) 형태로 통합하여 최종 위험도를 계산한다. 이는 사용자에게 단순한 AI 생성 여부가 아닌, 콘텐츠의 전반적 위험 수준을 명확히 전달한다. 이때의 최종 Risk Score는 가짜 정보 여부뿐 아니라 해당 콘텐츠가 사용자의 안전·프라이버시·심리적 피해에 미칠 수 있는 잠재적 위험을 함께 평가하도록 설계하였다.

위험도 계산에서 가장 중요한 요소는 각 위험 요소에 부여되는 가중치이다. 가중치는 다음과 같은 근거 기반 요소를 참고하여 설정한다.

1. 사회적 피해 규모
2. 과거 사례 및 통계
3. 플랫폼 정책

아래 표는 사용자 특성에 따라 상대적으로 중요한 위험 요소가 어떻게 달라질 수 있는지 보여주는 예시이다.

사용자	상대적 중요 위험 요소
여성	딥페이크, 디지털 성범죄
고령층	금융 사기, 의료 관련 허위 정보
청소년	폭력, 혐오

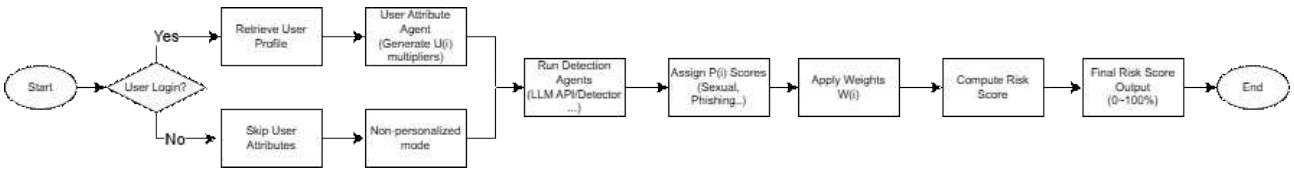
이와 같은 사용자 특성 기반 조정은 편향이 아닌 '사용자 보호 강화'를 목적에 하며, 동일한 콘텐츠라도 사용자 특성에 따라 체감·실질 위험도가 다르다는 점을 반영하여 보다 정교한 위험도 산출을 가능하게 한다.

또한 제안 엔진은 로그인 여부에 따라 두 가지 Risk Scoring 모드를 제공한다. 회원이 로그인한 경우, 성별·나이·직업 등 최소한의 사용자 프로필 정보를 수집, 활용하여 사용자 특성 기반 보정 계수 U(i)를 적용하고 개인화된 위험도를 산출한다.

비로그인 사용자의 경우, 개인정보를 제공하지 않으므로 모든 보정 계수 U(i)=1로 설정하여 콘텐츠 자체 분석 결과만을 반영한 非(비) 개인화 Risk Score를 제공한다. 이를 통해 개인정보를 제공하지 않는 비회원 사용자도 동일한 분석 기능을 사용할 수 있도록 하며, 동시에 로그인 사용자에게는 더욱 정밀한 맞춤형 위험 평가를 제공하는 선택적 개인화 구조를 구현하였다.

Detection Agents는 분석 대상 콘텐츠의 종류(텍스트·이미지·영상)에 따라 서로 다른 모델을 호출하며, 각 분석 결과는 통합된 위험 신호 벡터 P(i)로 정규화된다.

2) Risk score 계산을 통한 최종 위험 점수 도출



Risk score = $\sum W(i) \cdot P(i) \cdot U(i)$ (단, 비로그인 사용자의 경우 모든 U(i)=1 고정)

다음과 같은 가중합 형태의 선형 조합으로 계산된다. P(i)는 각 분석 모델에서 출력된 위험 점수(%), W(i)는 해당 위험 요소의 중요도를 반영한 가중치, U(i)는 사용자 특성 기반 보정 계수이다. 최종 Risk Score는 [0,1] 범위에서 산출된 정규화 값을 기반으로 0~100%의 백분율 형태로 변환되어 사용자에게 제공한다. 이를 통해 사용자는 위험 수준을 직관적으로 해석할 수 있다. 이 점수는 단순한 조작 여부 판정이 아니라, 콘텐츠 자체 위험도와 사용자 특성 기반 잠재적 피해 가능성을 함께 고려한 종합적 위험 평가 결과를 의미한다.

3) 문맥 기반 Risk 카테고리 분류 및 의사 결정 알고리즘 설계

① Risk 카테고리 분류

딥페이크와 AI 변조 기술은 목적에 따라 위험도가 크게 다르다. 아래 표와 같이 오락·성적·금전적 등 유형별 특성을 나누어 보면, 일부는 윤리적 문제 수준에 그치지만 성적·금전적 변조 등은 법적·사회적 파급력이 매우 크다. 따라서 이러한 차이를 반영한 명확한 리스크 분류와 이에 따른 정책적 대응이 필요하다.

카테고리	유형	리스크 높은 사람 유형
Fun	오락적, 유머적	- 유명인 - SNS 활동이 활발한 사람
Sexual	성적	- 유명인 - 개인적인 이미지가 온라인에 공개된 사람 - 여성, 아동
Deepfake /Deepface	얼굴 변조	- 정치인 - 유명 인플루언서 - 공공 인물
Phishing	금전적	- 일반 시민 - 금융 정보가 중요한 직장인

② 의사결정 알고리즘 설계

리스크는 아래와 같은 알고리즘으로 계산한다.

(1단계) 콘텐츠의 특성 및 목적 분석

카테고리	목적	리스크 높은 사람 유형
Fun	- 유머적 목적 - 장난, 친구들 간의 재미	- 얼굴 합성 - 이미지 변조 - GIF/비디오 생성
Sexual	- 성적 콘텐츠 생성 - 불법 성적 콘텐츠 제작	- 딥페이크 영상 생성 - 음성 합성 - 영상/이미지 변조
Deepfake/ Deepface	- 허위 정보 생성 - 명예훼손, 사회적 조작	- 얼굴 합성 - 딥러닝 모델 (GANs, Autoencoders) - 비디오 변조
Phishing	- 금전적 사기 - 개인정보 탈취	- 음성 합성 - 텍스트 음성 변환 (TTS) - 전화번호 합성

(2단계) 리스크 점수 할당

각 콘텐츠 유형에 대한 리스크 점수를 할당한다. 리스크 점수는 콘텐츠의 목적과 기술적 특성에 따라 다르게 부여된다.

위험도	리스크 점수	기준
Low Risk	0~0.3	사회적 영향이 적거나 법적 문제가 발생할 가능성이 낮은 콘텐츠
Moderate Risk	0.4~0.6	일정한 법적/사회적 영향을 미칠 수 있는 콘텐츠
High Risk	0.7~0.9	사회적 혼란을 초래하거나, 법적 문제를 발생시킬 수 있는 콘텐츠
Critical Risk	1.0	법적 조치와 사회적 대응이 필수적인 콘텐츠

(3단계) 리스크 점수 평가 및 대응 방안 마련

이후 할당된 리스크 점수를 기반으로 평가하고 리스크 수준을 결정한다. 리스크 점수는 합산하여 최종 리스크 수준을 도출한다. 점수에 따라 대응 방안을 설정하고, 리스크 수준에 따라 즉각적인 조치부터 법적 대응까지 상황에 맞는 대응이 필요하다.

위험도	리스크 점수	기준
Low Risk	0~0.3	추가 조치 없이 모니터링만 진행
Moderate Risk	0.4~0.6	경고 및 관리 필요, 모니터링 진행
High Risk	0.7~0.9	전문가 검토, 법적 대응 준비
Critical Risk	1.0 이상	즉각적인 법적 조치 및 사회적 대응 필수

4. 주요기능 및 기대효과

[주요기능]

본 시스템은 다층적 AI 구조를 기반으로 설계되었다. 특히 웹 기반 증강 특징을 고려한 텍스트 모델(ICNN-AEN-DM), LLM 기반 사실 검증 모듈, 이미지·딥페이크 탐지 모델, 외부 검색·팩트체크 API 등을 연동하여, 콘텐츠 단위의 종합적 신뢰도 평가를 수행한다. 아래 표와 같이 텍스트 분석·사실 검증·이미지 조작 탐지를 분리하여 처리함으로써, 단일 모델 기반 접근보다 더 높은 신뢰도와 설명력을 확보할 수 있다.

단계	에이전트/모듈	사용 AI 및 기술	주요 처리 내용

1. 입력 수집	Input Processing Agent	OCR / 크롤러 / 텍스트 파서	기사 URL·텍스트·이미지·영상 수집 및 전처리
2. 중앙 오케스트레이터	Orchestrator Agent	LLM / Knowledge Graph	각 에이전트 실행 조정·데이터 분배·결과 통합
3. 위·변조 (딥페이크) 탐지	Forgery Detection Agent	CNN 기반 이미지 조작 탐지 / 딥페이크 모델(Xception 등)	이미지·썸네일·프레임 기반 조작 여부 판별
4. 사실 검증 (Fact-check)	Verification Agent	LLM / Fact-check API / 검색엔진 API	주장 추출, 외부 근거와 비교, 사실 일치도 산출
5. 텍스트 기반 판별	Text Classification Agent	CNN + Autoencoder + MLP	본문 언어 패턴 분석 및 텍스트 기반 위험도 계산
6. 리스크 산출	Risk Scoring Agent	Rule-based + ML scoring	텍스트·이미지·사실성 등 종합 Risk Score 계산
7. 결과 출력	Reporting Agent	LLM / 구조화 템플릿	최종 가짜 뉴스 평가 보고서 생성 (High/Moderate/Low Risk)

또한 모든 에이전트는 중앙 오케스트레이터를 중심으로 상호 협력하며, 최종적으로 종합적인 가짜 정보 평가 보고서를 생성한다.

[비기능 요구]

1) 환경

- 사용자는 FAITH 서비스를 웹 브라우저에서 이용할 수 있다.
- 시스템은 백엔드 서버를 AWS EC2(Ubuntu 20.04 LTS 기반 Linux 환경)에서 운영하며, FastAPI·PostgreSQL을 포함한 서버 구성 요소가 정상적으로 동작할 수 있도록 표준화된 배포 환경을 유지한다.
- 클라이언트는 Next.js 기반 웹 환경에서 동작하며, 모든 주요 기능이 오류 없이 정상 작동하도록 설계한다.

2) 사용성

- 분석 결과(신뢰도, Risk score, 탐지 이유)는 텍스트 기반 설명을 함께 제공하여 비전문가도 이해할 수 있는 수준의 가독성을 보장한다.
- 사용자 대시보드는 반응형 UI로 제작하여 PC 환경에서 안정적인 사용 경험을 제공한다.
- 정보 접근성을 위해 API 응답 및 결과 화면은 간결한 문장과 근거 기반 설명의 형태로 제공한다.

3) 성능

- 시스템은 업로드된 콘텐츠(이미지·영상·텍스트)를 평균 2분 이내로 데이터 유형 분류 에이전트에 전달할 수 있어야 하며, 정상 처리율 90% 이상을 유지해야 한다.
- FastAPI 기반 서버는 비동기 처리를 이용해 AI 분석 API와의 통신 지연을 최소화하며, 동시 접속자의 요청을 안정적으로 처리해야 한다.
- PostgreSQL의 인덱싱 및 캐싱 구조를 활용해 검증 로그와 메타데이터 조회 시 1분 이내 응답을 목표로 한다.

4) 보안

- 사용자가 업로드하는 콘텐츠(이미지·영상·텍스트)는 모두 암호화된 통신(HTTPS)을 통해 서버로 전송되어야 하며 업로드된 콘텐츠는 기본값으로 비공개로 저장되고 사용자가 공개를 선택한 경우에만 외부에 노출될 수 있다.
- 관리자 기능(예: 외부 검증 API 키 등록, 내부 검증 모델 업데이트 등)은 관리자 권한이 부여된 계정에서만 접근할 수 있어야 한다.

[기대효과]

본 서비스 FAITH는 기술/사회/경제/교육 측면에서 다음과 같은 실질적 효과를 기대할 수 있다.

1) 기술적 효과

- 한국어 환경에 최적화된 AI 기반 팩트체크 기술을 확보함으로써, 국내 온라인 환경에 적합한 정밀한 검증 모델을 구축할 수 있다.
- 이를 통해 허위 정보 탐지 정확도를 향상하고 향후 다양한 도메인으로 확장할 수 있는 기술적 기반을 마련한다.

2) 사회적 효과

- 허위 정보 확산을 사전에 차단함으로써 디지털 정보 환경의 신뢰도를 제고한다.
- 이용자들은 더욱 안전하고 투명한 온라인 공간에서 소통할 수 있으며, 사회 전체의 정보 접근성과 공공성 증진에 기여한다.

3) 경제적 효과

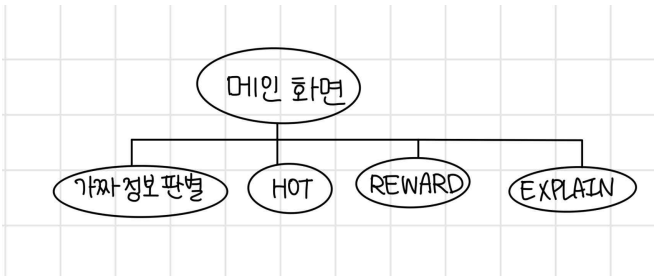
- 검증 데이터를 공공기관, 언론사, 플랫폼 기업 등과 연계함으로써 새로운 산업적 부가가치를 창출할 수 있다.
- 데이터 기반의 서비스 확장, API 제공 등으로 지속 가능한 비즈니스 모델로 발전할 가능성이 크다.

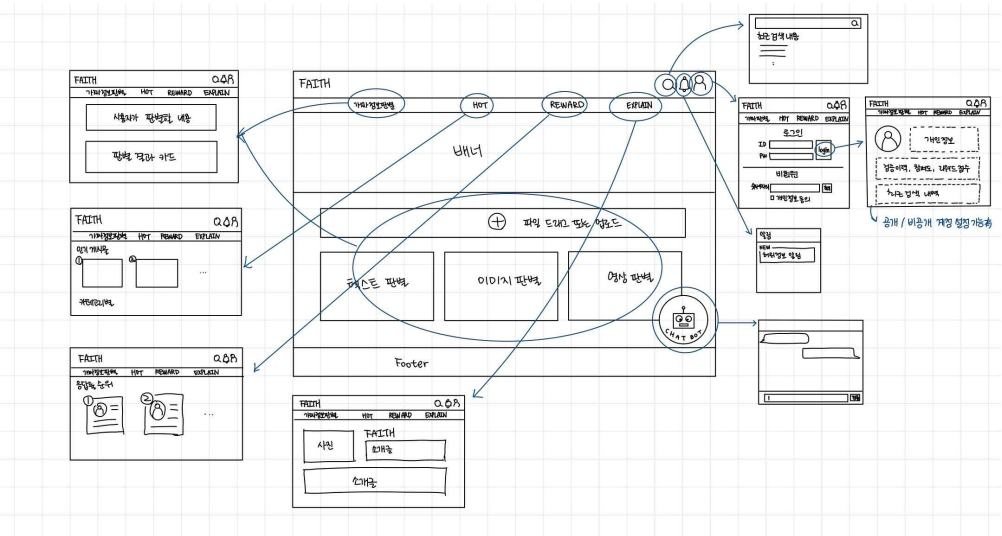
4) 교육적 효과

- 사용자 스스로 허위 정보를 판별하는 능력을 학습하도록 지원하여 미디어 리터러시 강화에 이바지한다.
- 이는 건전한 정보 이용 문화를 확산시키고, 장기적으로는 시민의 정보 판단 능력 향상이라는 사회적 자산을 형성한다.

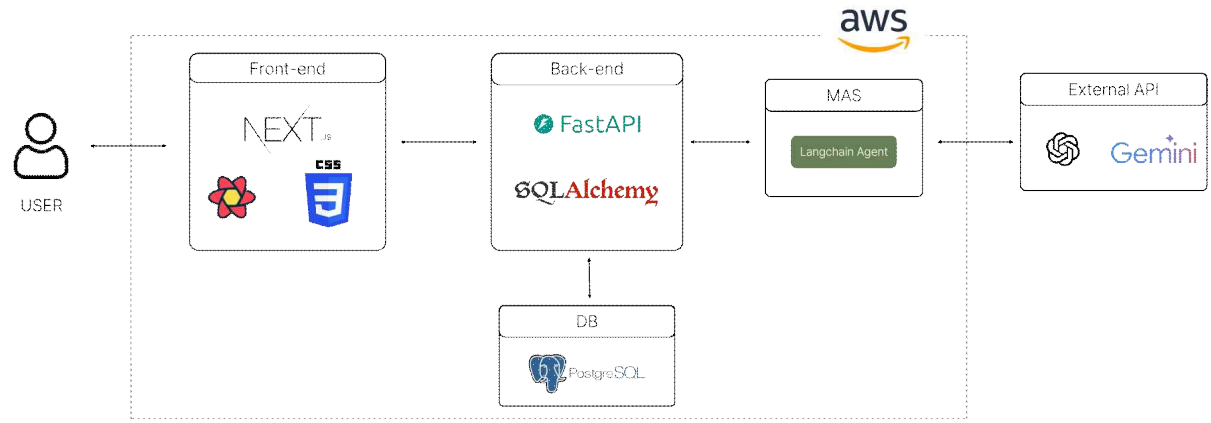
종합적으로, FAITH는 사용자 참여형/자가 학습 기반 플랫폼으로서 신뢰 중심의 정보 생태계를 구축하며, 기술/사회/경제/교육 전반에 걸쳐 지속 가능한 긍정적 변화를 이끌어낼 것이다.

[GUI 프로토타입]





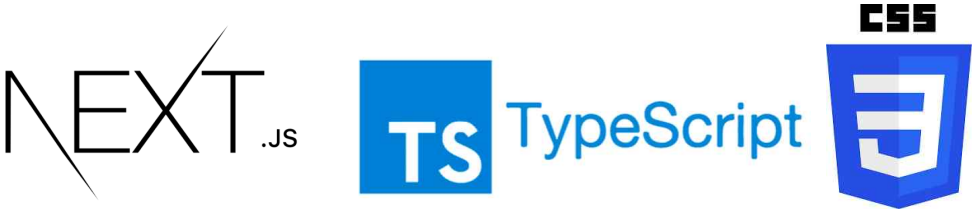
5. 개발 환경



본 프로젝트의 개발 및 서버 운영 환경은 Ubuntu 20.04 LTS 기반으로 구성된다. Ubuntu는 Linux 커널을 바탕으로 한 대표적 배포판으로, 안정성과 보안성이 높고 서버 운영에 최적화되어 있어 일관된 성능을 제공한다. 또한 광범위한 패키지 생태계와 오픈소스 라이브러리와의 높은 호환성을 바탕으로, 효율적인 개발·배포 환경을 구축할 수 있다.

백엔드 개발은 Python 3.x 기반으로 진행되며, 주요 웹 프레임워크로는 FastAPI를 사용한다. FastAPI는 Python 기반의 비동기 웹 프레임워크로, 높은 성능과 자동화된 API 문서화 기능을 제공한다. 이를 통해 백엔드 서버는 빠른 요청 처리 속도와 함께 RESTful API 구조로 안정적인 데이터 통신을 수행할 수 있다. 데이터베이스는 PostgreSQL을 활용하며, 관계형 구조를 기반으로 높은 안정성과 확장성을 제공한다. 또한 복잡한 연산이나 JSON 데이터 타입을 효율적으로 처리할 수 있어, 향후 데이터 분석 및 통계 활용에도 적합하다. 데이터베이스와의 상호작용은 SQLAlchemy ORM을 통해 수행한다. 이를 통해 SQL문 대신 Python 객체 단위로 데이터를 관리할 수 있어 코드의 가독성과 유지보수성을 높인다.

본 프로젝트는 MAS(Multi-Agent System) 구조를 채택하며, 에이전트 기반 로직 구현을 위해 LangChain Agent 프레임워크를 사용한다. 이를 통해 AI 모델 호출, 정보 검색, 판단 흐름 등을 모듈화하여 유연한 에이전트 아키텍처를 구성한다.



프론트엔드는 Next.js프레임워크를 사용한다. Next.js는 SSR과 SSG을 지원하여 초기 로딩 속도와 SEO를 개선한다. 또한 TypeScript를 함께 사용함으로써 정적 타입 검사를 통한 오류 방지와 코드 안정성을 확보하였다. 디자인 및 스타일링에는 CSS를 활용하여 반응형 UI를 구현하며, 데이터 시각화에는 ECharts와 Recharts를 적용하여 사용자 맞춤형 대시보드와 시각적 분석 기능을 제공한다.

상태 관리는 React-Query를 통해 API 요청 및 캐싱을 효율적으로 처리함으로써 UX를 강화한다.

버전 및 협업 관리는 GitHub을 기반으로 한다. Git을 이용해 코드 버전 이력을 관리하고, 브랜치 전략을 통해 병합 충돌을 최소화하며 협업 효율을 높인다. 모든 코드 변경 사항은 Pull Request 기반으로 검토되어, 품질 관리와 일관된 개발 프로세스를 유지한다.

배포 환경은 AWS 클라우드 플랫폼을 기반으로 구성한다. AWS EC2를 이용하여 서버를 운영하고, S3 및 RDS 서비스를 통해 파일 저장과 데이터 관리의 안정성을 강화한다. 클라우드 인프라를 통해 자동 확장성과 부하 분산이 가능하며, 서비스의 가용성과 신뢰성을 확보한다. 또한 향후 시스템 확장 시에도 유연한 대응이 가능하다.

AI 기능은 ChatGPT API와 Gemini API를 활용하여 구현한다. 사용자 입력 데이터를 기반으로 자연어 분석 및 사실 검증 기능을 수행하며, 이를 통해 프로젝트의 핵심 기능인 지능형 응답 시스템을 실현한다.

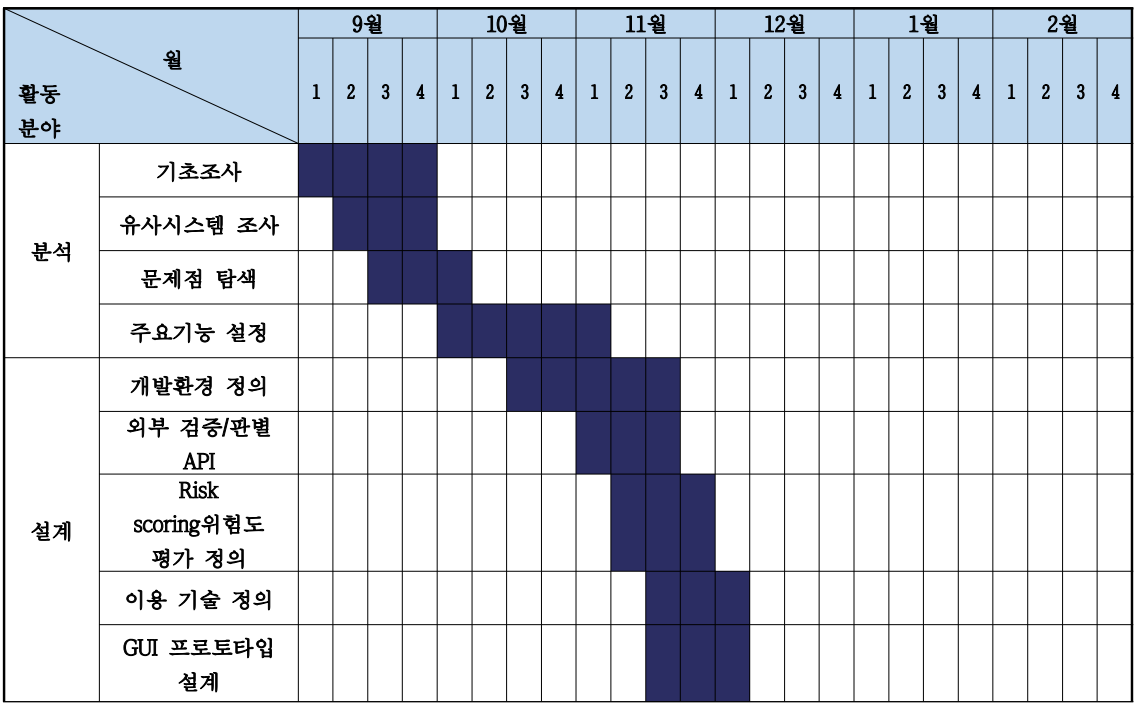
마지막으로, 클라이언트와 서버 간의 데이터 통신은 JSON 포맷을 기반으로 한다. RESTful 구조를 따르는 경량화된 포맷으로 시스템 간 데이터 교환의 효율성과 플랫폼 독립성을 동시에 확보한다.

6. 위험 요소

위험 구분	위험 요소	대응 전략	위험 수준
기술적 위험	외부 AI API 장애·지연	API 재시도 로직 적용, 타임아웃 설정, 장애 발생 시 대체 메시지 제공	high
데이터 위험	학습 데이터 부족·편향	공개 데이터셋 확장, 기사 데이터 크롤링, 다양한 소스의 데이터 통합, 판단 근거 제공	high
보안 위험	저작권 포함 콘텐츠 업로드	업로드 시 저작권 고지 문구 표시, 기본 비공개 처리, 신고·삭제 요청 기능 제공	high
운영 및 관리(통합) 위험	업데이트 중 서비스 중단 가능성	테스트 환경 운영, 점검 시간 사전 공지	moderate
사용자 위험	악의적 사용자의 오남용	업로드 횟수 제한, 비정상 패턴 자동 탐지	moderate

7. 일정 계획

- 25-2학기, 겨울방학



	통신 및 데이터											■	■	■										
	구조 정의												■	■	■									
	DB 모델링												■	■	■									
	위험요소 분석 및 전체 구조도 작성												■	■	■									
서버 처리부 구현	개발환경 구축													■	■	■								
	DB 스키마 작성														■	■	■							
	데이터 모델 구현														■	■	■							
	REST-API URL 정의 및 구현															■	■	■						
	보안 점검															■	■	■						
소비자 클라이언트 처리부 구현	개발환경 구축																■	■	■	■	■			
	레이아웃 작성																■	■	■	■	■			
	클라이언트 동작 구현																■	■	■	■	■			
개발	개발환경 구축																■	■	■					
서버 연동	REST API 연동																	■	■	■	■	■		
	실제 데이터 처리 동작 구현																				■	■	■	■

- 26-1학기

활동 분야 \ 월		3월				4월				5월	6월
		1	2	3	4	1	2	3	4		
테스트	디버깅 및 테스트	■	■	■	■	■					
유지보수	유지보수			■	■	■	■				
보고서	보고서 작성					■	■	■	■		

8. 참고 문헌

[1] 전하연,「팩트체크 뉴스의 검증 방식과 판정 결과 일치 여부가 이용자의 판단 변화와 판단에 대한 확신 정도에 미치는 영향」, 서울대학교 대학원 석•박사학위논문, 2024, P18.

[2] 최순옥&윤석민,「협업형 사실검증 서비스의 의의와 과제: <SNU팩트체크>의 사례」, 2017, P33.

[3] 오세욱&황구현,「'팩트'의 형식적 구성 요건에 대한 탐색적 연구」, 2018, P12.

[4] Jae-Seung Shim&Ha-Ram Won&Hyunchui Ahn,「A Study on the Effect of the Document Summarization Technique on the Fake News Detection Model」, 2019, P1.

[5] 김고은, "'재정 고갈' SNU팩트체크, 무기한 운영 중단", 한국기자협회, 2024. 08. 19, https://www.journalist.or.kr/m/m_article.html?no=56539&utm_source

[6] 김고은, "기자와 시민 협력해 허위조작정보 맞서는 '팩트체크넷' 출범", 한국기자협회, 2020. 11. 12, https://www.journalist.or.kr/news/article.html?no=48438

[7] 금준경, "문재인 정부 때 만든 시민 참여 팩트체크 서비스 '중단'", 30미디어오늘, https://www.mediatoday.co.kr/news/articleView.html?idxno=308429

[8] 성재호,「시민과 함께 하는 '팩트체크넷'이 출범합니다」, 2020, P94-95.

[9] 이홍천,「영미일 언론사들의 팩트체크 사이트 분석 한국 언론에 대한 시사점」, 2024, P3.

[10] SOU HYUN JANG&KYOUNG EUN JUNG&YONG JEONG YI,「The Power of Fake News: Big Data Analysis of Discourse about COVID-19–Related Fake News in South Korea」, 2023, P2.

[11] Project Manager Template, "What Is a Risk Category? Effective Risk Management", 2023, from https://www.projectmanagertemplate.com/post/what-is-a-risk-category-effective-risk-management)

[12] Massachusetts Institute of Technology, "Risk Classifications", n.d., from https://infoprotect.mit.edu/risk-classifications/

[13] Akira AI, "Agentic AI for Fraud Prevention: Transforming Risk Detection with Autonomous Agents", 2024, from https://www.akira.ai/blog/agentic-ai-for-fraud-prevention

[14] Ali, A.M.; Ghaleb, F.A.; Mohammed,M.S.; Alsolami, F.J.; Khan, "Web-Informed Augmented Fake News Detection Model Using Stacked Layers of Convolutional Neural Network and Deep Autoencoder". Mathematics 2023, 11, 1992. from https://doi.org/10.3390/math11091992

[15] Akira AI, "Risk Management with Akira AI: How Intelligent Agents Enhance Decision-Making", 2024, from https://www.akira.ai/blog/risk-management-with-akira-ai