**문맥 기반 AI 안내 시스템 기술**

**로고, 상징, 엠블럼, 등록 상표이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

|  |
| --- |
| **담당 함형민 교수님** |
| **제출일 2025년 7월 30일** |
| **성명: 한승연,유민석,유제이,오지윤** |

**목차**

**Ⅰ. Executive Summary (요약)**

* 제안서의 목적 및 배경
* 시스템의 필요성과 기대효과
* 제안 시스템 개요

**Ⅱ. Problem Statement (문제 정의)**

* 현 시스템의 문제점 및 한계점 분석
* 개선해야 할 기술적/운영적 문제점

**Ⅲ. Solution Overview (솔루션 개요)**

* 제안하는 시스템의 개념 및 장점
* 시스템 목표 및 기대성과

**Ⅳ. 기술 요약 (Technical Summary)**

* 시스템의 핵심 기술 소개
* 문맥 기반 AI 안내 시스템의 필요성 및 특징
* 시스템 구성 요소 및 기술 스택 개요

**Ⅴ. 시스템 설계 (System Design)**

* **시스템 구조 및 설계 개요**
* **AI 계층 전처리·동작 단계(AI Layer)**
* **데이터 관리 계층 저장·관리 방안 (Data Management Layer)**
* **데이터 흐름 및 처리 시나리오**

**Ⅶ. 배포 및 통합 (Deployment and Integration)**

* **시스템 배포 계획 및 방법**
  + 클라우드 기반 인프라 구축 방안
  + 통합 및 연동 방안 설명
* 기존 시스템과의 호환성 및 데이터 이관 계획
* 유지보수 및 확장성 전략

**Ⅸ. 결론 (Conclusion)**

* 제안한 시스템의 기대 효과 및 장기적 비전

## **Ⅰ. Executive Summary (요약)**

**Ⅰ-1. 제안서의 목적 및 배경**

 본 제안서의 목적은 사용자의 웹사이트 탐색 경험을 향상시키기 위해 최신 문맥 기반 AI 기술과 고급 UI 오버레이 시스템을 활용하여 효율적인 안내 시스템을 제공하는 것이다.

 기존의 안내 방식은 고정적이고 사용자의 문맥을 반영하지 못하여 사용자가 원하는 정보를 즉시 찾기 어려웠으며, 이는 사용자의 서비스 만족도와 생산성 저하로 이어졌다.

 따라서, 본 제안에서는 자연어 처리, 문맥 이해, 사용자 행동 분석 등의 첨단 AI 기술과 유연하고 비침투적인 UI 오버레이를 통해 개인 맞춤형 안내를 제공하여 이러한 문제를 해결하고자 한다.

**Ⅰ-2. 시스템의 필요성과 기대효과**

 원본 시스템은 자연어 처리(NLP)와 문맥 예측 기술(spaCy, KoBERT, Transformers)을 활용하여 사용자의 의도를 정확히 파악하고, 이를 통해 최적의 탐색 경로를 제시함으로써 탐색 시간을 획기적으로 단축할 수 있다.

 또한, FAISS, Pinecone 등 유사도 검색 엔진을 통해 사용자에게 가장 적합한 안내 경로를 실시간으로 추천하고, TensorFlow를 이용한 행동 기반 분석을 통해 사용자의 반복적이고 비효율적인 탐색 패턴을 최소화할 것으로 예상된다.

**Ⅰ-3. 제안 시스템 개요**

제안하는 시스템은 크게 AI 계층, UI 오버레이 계층, 시스템 통합 및 배포 계층으로 구성된다.

 AI계층: 의미 추론(KoBERT, mT5), 유사도 탐색(FAISS, Pinecone), 사용자 행동 분석(TensorFlow.js),

키워드 연산(NLP-Cube, WordNet)을 통해 정확한 문맥 분석과 맞춤형 안내를 제공한다.

 UI 오버레이 계층: React Portal, Shadow DOM을 이용한 비침투적 UI 삽입과 GSAP, Framer Motion을 통한 직관적인 시각 안내 요소로 사용자에게 원활한 안내 경험을 제공한다.

 통합 계층에서는 Redux, Recoil 등을 사용한 상태 관리와 Chrome Extension, Electron을 통한 유연한 배포 방식을 제공하여 다양한 환경에 쉽게 적용할 수 있도록 설계한다.

## **Ⅱ. Problem Statement (문제 정의)**

**Ⅱ-1. 현 시스템의 문제점 및 한계점 분석**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.본 그래프는 사용자가 현재 시스템 내에서 제공되는 주요 5가지 기능(Feature)에 대해 인지(Awareness)하고 있는 정도와 실제로 활용(Usefulness)하고 있는 정도의 차이를 나타낸 것이다.

분석 결과, 모든 기능에서 사용자의 인지 수준과 실제 활용도 간에 유의미한 차이가 존재하는 것으로 나타났다. 특히 Feature 2의 경우 사용자의 인지 수준은 50%에 불과하였으나, 실제 활용도는 92%로 큰 격차를 보였다. 반대로 Feature 5는 사용자의 인지도가 가장 낮은 37%로 나타났으나, 활용도는 79%로 상대적으로 높은 편이었다.

이러한 결과는 현 시스템이 제공하는 기능 자체의 유용성은 높으나, 사용자가 해당 기능의 존재를 인식하지 못하거나 발견하기 어려워 제대로 활용하지 못하고 있음을 시사한다. 즉, 시스템의 문제점은 사용자에게 필요한 기능이 명확하게 노출되지 않고 있으며, 기능 안내와 사용자 인터페이스(UI)의 설계가 직관적이지 않다는 데 있다.

이에 따라 본 시스템의 한계점으로는 다음과 같은 사항들이 지적될 수 있다.

|  |  |
| --- | --- |
| 구분 | 문제점 및 한계점 |
| 접근성 및 가시성 | 주요 기능에 대한 접근성과 가시성이 떨어져 사용자의 기능 인지가 부족하다. |
| 연결성 부족 | 기능 간 연결성이 미흡하여 사용자가 필요한 기능을 찾는 과정에서 혼란을 겪고 있다. |
| 탐색 시간 증가 | 사용자가 원하는 정보를 탐색하는 데 소요되는 시간이 길어지며, 이는 서비스 만족도 및 사용자 경험(UX)을 저하시키는 요인으로 작용하고 있다. |

**Ⅱ-2. 개선해야 할 기술적/운영적 문제점**

현 시스템에서 나타난 문제점과 한계점은 사용자 경험을 크게 저하시키고, 서비스 효율성에도 부정적인 영향을 미친다. 따라서 사용자 중심의 개선 방안이 반드시 필요하며, 이를 위해 다음과 같은 구체적인 개선이 요구된다.

|  |  |
| --- | --- |
| 구분 | 개선 방안 |
| 접근성 및 가시성 향상 | UI 디자인 개선 및 명확한 안내 시스템 구축 |
| 기능 연결성 강화 | 사용자가 기능을 보다 쉽게 찾고 이해할 수 있도록 기능 간의 논리적 연결성 강화 |
| 탐색 시간 단축 | 문맥 기반의 개인 맞춤형 안내 시스템 도입 |

이러한 개선을 통해 사용자 만족도를 높이고 서비스의 전반적인 품질과 효율성을 향상시킬 수 있다.

## **Ⅲ. Solution Overview (솔루션 개요)**

**Ⅲ-1. 제안하는 시스템의 개념 및 장점**

제안 시스템은 최신 인공지능(AI) 기술과 사용자 중심의 인터페이스(UI)를 융합하여, 사용자의 웹사이트 탐색 과정에서 발생하는 접근성 문제와 기능 인지 부족 현상을 개선하기 위한 솔루션이다. 특히, 본 시스템은 문맥 기반 개인 맞춤형 안내 시스템을 통해 사용자의 목적과 의도에 맞는 정보를 정확하고 신속하게 제공하여 탐색 효율성을 극대화하고자 한다.

|  |  |
| --- | --- |
| 구분 | 장점 |
| 개인 맞춤형 안내 | 사용자의 문맥과 사용 패턴을 실시간 분석하여 개별 사용자의 목적에 맞는 정보를 제공 |
| 높은 접근성 및 가시성 | 직관적이고 명확한 UI 디자인을 적용하여 사용자들이 기능과 정보를 쉽게 인지하고 접근할 수 있도록 지원 |
| 기능 간 논리적 연결성 강화 | 각 기능 간의 관계를 명확히 하여 사용자들이 원하는 기능을 직관적으로 찾고 이해할 수 있도록 지원 |
| 탐색 시간 단축 | 사용자의 맥락에 따라 최적의 경로를 제시함으로써 탐색에 소요되는 시간을 획기적으로 단축 |

**Ⅲ-2. 시스템 목표 및 기대성과**

본 제안 시스템의 목표는 다음과 같다.

* 웹사이트 기능 및 메뉴의 접근성과 가시성 향상
* 사용자 맞춤형 정보 제공을 통해 사용자 만족도를 높인다.
* 탐색 시간을 단축하여 서비스 이용의 효율성 극대화.

|  |  |
| --- | --- |
| 기대성과 | 내용 |
| 사용자 경험 개선 | 사용자 중심의 맞춤 안내를 통해 원하는 정보를 쉽게 찾고 효율적으로 탐색할 수 있어 전반적인 사용자 경험이 향상됨 |
| 서비스 이용률 향상 | 시스템 이용의 편의성과 효율성이 증대됨에 따라 사용자의 재방문율과 전반적인 서비스 사용률이 증가 |
| 서비스 운영 효율화 | 탐색 과정의 효율화를 통해 사용자 지원 및 안내에 필요한 인적·물적 자원을 절약하고 효율적으로 운영 가능 |

## **Ⅳ. 기술 요약 (Technical Summary)**

본 기술 요약에서는 제안하는 시스템을 구성하는 핵심 기술, 문맥 기반 AI 안내 시스템의 필요성과 특징, 시스템의 구성 요소 및 사용되는 기술 스택을 개괄적으로 소개한다.

**Ⅳ-1. 시스템의 핵심 기술 소개**

* **AI 기반 문맥 분석**: 사용자의 입력, 행동 데이터, 페이지 문맥을 심층적으로 해석하여 사용자의 의도와 요구를 정확하게 파악한다.
* **직관적인 UI 오버레이 구현:** 웹 사이트 기본 구조를 침해하지 않으면서 직관적인 UI 오버레이를 제공하여 사용자가 자연스럽게 안내를 받을 수 있도록 한다.
* 유연한 시스템 통합 및 배포 환경: 다양한 환경(웹, 크롬 확장, 데스크톱 등)에 쉽게 적용할 수 있도록 통합성과 확장성이 뛰어난 구조를 제공한다.

**Ⅳ-2. 문맥 기반 AI 안내 시스템의 필요성 및 특징**

|  |  |
| --- | --- |
| 구분 | 내용 |
| 필요성 | 기존 웹사이트의 메뉴나 정보 구조가 복잡해질수록 사용자가 원하는 정보를 빠르게 찾기 어려워진다. 문맥 기반 AI 안내 시스템은 이러한 문제를 해결하기 위해 사용자의 실제 목적과 상황을 실시간으로 파악, 맞춤형 안내를 제공한다. |
| 특징1 | 의미 추론(KoBERT, mT5)과 사용자 행동 분석(TensorFlow.js 등)을 결합하여 사용자의 니즈와 상황을 신속하게 해석한다. |
| 특징2 | 유사도 탐색 및 키워드 연산 등 다양한 AI 모듈을 활용해 개인별로 최적화된 정보 제공한다. |
| 특징3 | 비침투적 UI 오버레이와 결합되어, 기존 시스템을 크게 변경하지 않고도 적용 가능하다. |

**Ⅳ-3. 시스템 구성 요소 및 기술 스택 개요**

시스템의 구성 요소 및 기술 스택은 다음 표와 같다.

|  |  |
| --- | --- |
| 계층 | 사용 기술 |
| AI 계층 | 의미 추론(KoBERT, mT5), 유사도 탐색(FAISS, Pinecone), 사용자 행동 분석(TensorFlow.js), 키워드 연산(NLP-Cube, WordNet) |
| UI 오버레이 계층 | React Portal, Shadow DOM, GSAP, Framer Motion |
| 시스템 통합 및 배포 계층 | Redux, Recoil, Chrome Extension, Electron |

각 계층별로 도입된 기술은 문맥 분석, 사용자 행동 예측, 인터페이스 가시성 강화, 다양한 환경에서의 유연한 통합·배포 등 시스템의 전 과정을 유기적으로 연결한다. 이를 통해 사용자는 기존 웹사이트 구조의 변화 없이, 효율적이고 직관적인 맞춤형 안내 서비스를 경험할 수 있으며, 운영자는 다양한 플랫폼 환경에서도 일관된 서비스 품질을 유지할 수 있다.

## **Ⅴ. 시스템 설계 (System Design)**

**Ⅴ‑1. 시스템 구조 및 설계 개요**

본 시스템은 **사용자 브라우저(Client) → API Gateway → AI 계층 → 데이터 관리 계층**으로 이어지는 4‑계층 전체 아키텍처 흐름은 아래 그림과 같이 구성된다.

텍스트, 도표, 스크린샷, 디자인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**Ⅴ‑1-1. 아키텍처 흐름도**

|  |  |  |
| --- | --- | --- |
| 계층 | 핵심 구성요소 | 데이터 흐름 요약 |
| Client | UI Overlay Layer (React Portal, Shadow DOM, GSAP/FramerMotion) Behavior Logger SDK (PostHog / Segment) | 사용자 이벤트를 REST 또는 WebSocket으로 전송 |
| API Gateway | NestJS / FastAPI 인증 · 라우팅 · 속도 제한 | |  |  | | --- | --- | |  | 이벤트를 AI 계층으로 전달 | |
| AI 계층 | NLP Engine ▶ Similarity Search ▶ Context‑Inference ▶ Recommendation Orchestrator Log Processor (Kafka Topic → Flink/Spark) | 실시간 분석 결과를 WebSocket으로 즉시 안내하고,스트리밍 로그를 데이터 계층으로 전송 |
| 데이터 관리 계층 | PostgreSQL, TimescaleDB, S3 Blob, Feature Store | 로그·feature 저장 및 모델 재학습 파이프라인 |

**Ⅴ‑1-2. 주요 흐름**

주요 흐름은 다음과 같다.

텍스트, 스크린샷, 도표, 만화 영화이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

1. **이벤트 수집**: 브라우저에서 Logger SDK가 클릭·스크롤 이벤트를 전송
2. **게이트웨이 처리**: 인증·라우팅 후 AI 계층으로 전달
3. **AI 추론**: NLP → 유사도 → 컨텍스트 → 추천 오케스트레이션
4. **실시간 안내**: WebSocket으로 힌트/프롬프트 송신
5. **로그 적재 & 재학습**: Kafka → Flink/Spark → Feature Store → Airflow/MLflow

**Ⅴ‑2. AI 계층 (AI Layer)**

아래 표는 AI 계층이 사용자 입력을 받아 NLP 처리부터 추천·로그 적재까지 수행되는 전체 파이프라인을 단계별로 정리한 것이다. 각 열은 입력, 알고리즘, 산출물, 그리고 사용된 핵심 기술 스택을 한눈에 보여준다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 단계 | 주요 입력 | 처리 / 알고리즘 | 출력 | 관련 모듈·기술 |
| 1. NLP 전처리 | 사용자 자연어 입력, 페이지 컨텍스트 | 토큰화 → 임베딩(Transformer) → 의미 추론 | 토큰 벡터, 키워드 리스트 | NLP Engine (KoBERT, mT5, spaCy) |
| 2. 유사도 검색 | 토큰 벡터, 키워드 | 벡터 DB Nearest‑Neighbor 검색 | 후보 UI 경로 IDs / Paths | Similarity Search (FAISS, Pinecone) |
| 3. 컨텍스트 추론 | 후보 경로, 행동 로그(Feature) | Lostness 모델, Next‑Step 예측 모델 | 길잃음 점수, 다음 클릭 확률, 프로필 스코어 | Context‑Inference & User‑Model (TF.js, ONNX.js, LightGBM) |
| 4. 추천 조합 | 컨텍스트 결과, 후보 경로 | 비즈니스 룰 + 가중치 최적화 | 안내 문구, 말풍선 위치, 애니메이션 타입 | Recommendation Orchestrator (Node.js, Go) |
| 5. 실시간 전송 | 안내 페이로드 | WebSocket 브로드캐스트 | 브라우저 UI 오버레이 표시 | Recommendation Orchestrator |
| 6. 로그 스트리밍 | 원시 행동 로그, 추론 결과 | Kafka Producer → Flink/Spark 변환 | 정규화 Feature, 지표 집계 | Streaming Logs & Log Processor |
| 7. Feature Store 적재 | 정규화 Feature | Batch ETL, Schema Mapping | 최신 Feature Store 업데이트 | Flink/Spark ETL, Kafka → S3 / DB |

**Ⅴ‑3. 데이터 관리 계층 저장·관리 방안**

|  |  |  |
| --- | --- | --- |
| 저장소 | 주요 내용 | 활용 목적 |
| PostgreSQL | 사용자 프로필, 시스템 설정, 권한 메타데이터 | 트랜잭션성 데이터 저장 |
| TimescaleDB | 클릭·스크롤 등 시계열 행동 로그 | 실시간 대시보드, 행동 패턴 분석 |
| S3 Blob Storage | 세션 레코드, 스크린샷, 모델 Artifact | 모델 재학습, 감사(Compliance) |
| Feature Store | ETL로 생성된 모델 입력용 Feature | 온라인·오프라인 학습 공용 |

본 표는 데이터 관리 계층이 각기 다른 특성을 가진 저장소를 활용하여 로그·Feature·모델 아티팩트 등을 체계적으로 보관하고 재학습 및 분석에 활용하는 구조를 요약한 것이다.

**Ⅴ‑4. 데이터 흐름 및 처리 시나리오**

본 시스템의 데이터는 실시간 안내, 스트리밍 학습, 배치 리포트의 세 가지 시나리오를 통해 순환‑처리된다. 각 시나리오는 사용자 상호작용부터 모델 재학습·리포트 생성까지 일관된 데이터 흐름을 유지해 실시간성(Reactivity), 확장성(Scale‑out), 데이터 일관성(Consistency)을 동시에 달성한다.

|  |  |  |  |
| --- | --- | --- | --- |
| 데이터 흐름 유형 | 주요 소스 | 처리 경로 | 목적 / 최종 산출물 |
| 실시간 안내 흐름 | 브라우저(클릭·스크롤 이벤트) | REST → API Gateway → NLP Engine → Similarity Search → Context Inference → Recommendation Orchestrator → WebSocket → 브라우저 | 화면 오버레이로 실시간 힌트·말풍선 표시 |
| 스트리밍 학습 흐름 | 브라우저 행동 로그 | Kafka Topic → Log Processor(Flink/Spark) → Feature Store 업데이트 → 모델 재학습(Airflow/MLflow) | 최신 사용자 행동을 반영한 AI 모델 성능 향상 |
| 배치 리포트 흐름 | TimescaleDB 시계열 로그 | Spark 배치 작업 → BI 대시보드 | |  | | --- | |  |   주/월간 행동 패턴 리포트   |  | | --- | |  | |

이와 같은 모듈형 설계 덕분에 각 흐름은 서로 영향을 주지 않으면서도 데이터 파이프라인을 재사용하여 운영 효율과 안정성을 극대화할 수 있다.

## **Ⅶ. 배포 및 통합 (Deployment and Integration)**

**Ⅶ‑1. 시스템 배포 계획 및 방법**

* 클라우드 기반 인프라 구축: AWS EKS(쿠버네티스) + S3 + RDS (PostgreSQL) 조합으로 서비스 노드, AI 모델 서버를 컨테이너 단위로 오토스케일링.
* 하이브리드 배포 모델: 웹 플러그인(Chrome Extension)·UserScript로 기존 사이트에 삽입하거나, Electron 패키징으로 데스크톱 독립 앱 형태로 배포 가능.
* CI/CD 파이프라인: GitHub Actions → Docker Registry → EKS Rolling Update.

**Ⅶ‑2. 통합 및 배포 계층 기술 구성**

|  |  |  |
| --- | --- | --- |
| 분류 | 적용 기술 | 설명 |
| 사이트 삽입 / 배포 | Chrome Extension, UserScript (Tampermonkey), Electron | 기존 사이트 주입, 확장 기능, 데스크톱 앱 등 다중 채널 배포 지원 |
| 통합 프레임워크 | Next.js, Vue 3, SvelteKit, Vite | UI·AI·오버레이 상태를 통합 관리하는 SSR/SPA 프레임워크 |
| 상태 관리 | Redux, Recoil, Vuex | 오버레이 위치·표시 여부·AI 결과를 전역 상태로 공유 |
| 시나리오 테스트 / UI 검증 | Storybook, Jest, Playwright, Cypress | 컴포넌트·흐름 단위 자동 테스트·시각 회귀 테스트 |
| 보안 & 권한 처리 | CSP 우회, iframe sandbox, DOM 접근 제한 우회 | 3rd‑party 사이트에 오버레이 삽입 시 보안 정책 대응 |

**Ⅶ‑3. 기존 시스템 호환성 및 데이터 이관 계획**

1. **점진적 롤아웃**: 기존 서비스에 Chrome Extension 방식을 우선 적용 → 안정성 검증 후 본 서버 통합 배포.

2. **데이터 마이그레이션**: 기존 DB 스키마를 PostgreSQL 로 옮기고, ETL 스크립트로 TimescaleDB 시계열 로그 테이블 분리.

3. **백엔드 호환 API:** 기존 REST 엔드포인트와 동일한 URI 패턴을 API Gateway에 프록시 매핑해 레거시 클라이언트 중단 없이 전환.

**Ⅶ‑4. 유지보수 및 확장성 전략**

* 모놀리스↔마이크로서비스 전환 유연성: 오케스트레이터, NLP, Similarity Search 등을 독립 컨테이너로 분할 배치 가능.
* 오토스케일링: EKS HPA + KEDA로 AI 추론 노드의 CPU/GPU 사용량 기반 자동 확장.
* Observability: Prometheus/Grafana 대시보드, Loki 로그 스택으로 실시간 모니터링.
* 플러그인 생태계: 오버레이 UI 컴포넌트를 Web Component 기반으로 설계해 신규 사이트·테마에 쉽게 확장.

## Ⅸ. 결론 (Conclusion)

본 제안 시스템은 AI 기반 문맥 분석과 비침투적 UI 오버레이를 결합하여, 사용자가 웹사이트에서 원하는 정보를 빠르고 직관적으로 찾을 수 있도록 지원한다. 이를 통해 사용자 경험(UX)을 혁신적으로 개선하고, 서비스 운영 효율성을 극대화할 수 있다.

|  |  |  |
| --- | --- | --- |
| 구분 | 기대 효과 | 상세 내용 |
| 사용자 경험 향상 | 탐색 시간 단축 · 길잃음 해소 | 실시간 맞춤형 안내로 평균 탐색 시간을 대폭 감소시키고, 사용자 만족도·재방문율 상승 |
| 서비스 품질 향상 | 개인화·정확도 증대 | 행동 로그 기반 AI 모델이 지속 학습하여 추천 정확도·UI 안내 품질이 점진적으로 향상 |
| 운영 효율성 | 지원 비용 절감 | 문의·헬프데스크 부담 완화, 내부 가이드·FAQ 작성 비용 절감 |
| 데이터 자산 확보 | 행동 로그·Feature 축적 | 체계적 데이터 수집·관리로 신규 AI 기능(예: 예측 분석) 개발 기반 확보 |
| 확장성 & 유연성 | 멀티 채널·멀티 도메인 적용 | Chrome Extension, Electron 앱 등 다양한 배포 채널과 레거시 시스템 호환 |

**장기적 비전**

**• API 개방형 AI 서비스 플랫폼:** Recommendation Orchestrator와 NLP 모듈을 SaaS 형태로 외부에 공개하여 파트너 서비스도 손쉽게 통합할 수 있는 에코시스템 조성

**• 완전 자동화된 MLOps 파이프라인:** Airflow·MLflow 기반으로 모델 학습·검증·배포를 자동화하고, 실시간 모니터링까지 포함해 지속적인 모델 품질 보장 • 산업·도메인 확장: 전자상거래, 교육, 공공 포털 등 다양한 산업으로 적용 범위를 넓혀 맞춤형 안내·추천 시장 선도

**• 데이터 거버넌스 고도화:** Feature Store와 로그 데이터에 대한 보안·프라이버시·품질 정책을 표준화하여 기업 전사 데이터 거버넌스 체계 구축

이와 같은 단계적 확장 로드맵을 통해 본 시스템은 단순 내비게이션 지원 솔루션을 넘어, 데이터 기반 디지털 경험 최적화 플랫폼으로 성장할 수 있을 것이다.