

[Personal] NewsPulse

≡ 프로젝트 주최	본인
≡ 규모	1인
≡ 날짜 및 기간	2024-11~2025-01
≡ 사용 기 술	Crawling Docker EC2 FastAPI SQLAlchemy View langchain postgresql
≡ 설명	관심분야 뉴스 시각화 대시보드

NewsPulse

뉴스 검색...

웃는남자

르브론

LLM

실시간 트렌드

카테고리별 뉴스 발행 추이

데이터가 없습니다

실시간 뉴스

최신순

르브론

2024-12-10T02:13:32

세계 최고 계약 '765M 소토 > NFL No.1+NBA No.2'

르브론

2024-12-09T08:13:32

[NBA] 르브론 없는 레이커스, 역시 AD가 왕이었다...2G 연속 30점 이상 과력

르브론

2024-12-08T08:13:32

최근 9경기에서 7패→르브론 "방법 찾기 위해 노력하지만 어려운 상황"

르브론

2024-12-07T08:13:32

트리플더블 르브론의 반등+AD의 강력함. LA 레이커스 그라운드 3연패, 서부 10위 위기

르브론

2024-12-07T08:13:32

NBA 애클랜드, '제임스 트리플더블' 레이커스 꺾고 6연승

르브론

2024-12-06T08:13:32

"부끄럽다", 41점 차 대패 후 '킹' 르브론이 남긴 한 ntr!

관련 링크

<https://github.com/JooyoungOfficielEpitech/NewsPulse>

[뉴스펄스 홈페이지](#)

프로젝트 개요

NewsPulse는 정보가 넘치는 지금, 사용자의 관심사에 따라 여러 플랫폼 (뉴스, 커뮤니티 등등) 에서 데이터를 불러와 한눈에 현황을 모니터링하는 웹앱을 제공합니다.

상세 내용

- **설명:** 사용자 관심 키워드에 따라 뉴스 데이터를 크롤링, 분석하여 트렌드 데이터를 생성 및 제공하는 시스템 개발. 프론트엔드와 백엔드를 모두 개발하며, 사용자 경험을 고려한 인터페이스와 API를 설계.
 - **사용 대상:** 뉴스 트렌드 분석을 통해 정보를 활용하려는 사용자.
 - **핵심 기능:**
 - 사용자 카테고리 및 키워드 관리
 - 뉴스 데이터 크롤링 및 저장
 - 뉴스 제목 및 내용 기반 트렌드 분석
 - 실시간 데이터 조회 및 뉴스 내용을 기반한 챗봇 제공
 - React 기반의 직관적인 사용자 인터페이스 제공
 - 백엔드 API와 프론트엔드 간 원활한 데이터 통신
-

사용 기술 및 라이브러리

- **Frontend:** React, Axios, Chart.js, React Query, Tailwind CSS
 - **Backend:** FastAPI, SQLAlchemy, Asyncio
 - **Database:** PostgreSQL
 - **Scheduler:** APScheduler (Python)
 - **ETC:**
 - Github CI/CD
 - BeautifulSoup (크롤링)
 - konlpy (형태소 분석)
 - Docker (배포 환경 구성)
 - AWS EC2 (서버 호스팅)
 - Langchain / OpenAI (챗봇)
-

본인 참여 기능

프론트엔드

- **React 기반 UI/UX 개발:**
 - 사용자 카테고리 및 키워드 관리 페이지 구현.
 - 트렌드 시각화 및 데이터 표출을 위한 인터페이스 개발.
 - React Query를 활용한 비동기 데이터 요청 최적화.
 - Chart.js를 사용하여 트렌드 데이터를 시각화.
- **스타일링 및 반응형 디자인:**
 - Tailwind CSS를 사용한 간결하고 유지보수 가능한 스타일링.
 - 모바일과 PC에 최적화된 반응형 디자인 구현.
- **API 연동:**
 - Axios를 사용하여 백엔드와의 데이터 통신 구현.
 - API 오류 처리 및 로딩 상태 관리.

백엔드

- **API 개발:**
 - FastAPI로 RESTful API 설계 및 구현.
 - 사용자 카테고리 및 키워드 CRUD API 개발.
 - 뉴스 데이터 크롤링 및 트렌드 분석 API 구현.
- **데이터 분석 및 저장:**
 - PostgreSQL에 크롤링된 뉴스 데이터 저장 및 쿼리 최적화.
 - 트렌드 분석 알고리즘 개발 및 저장 로직 구현.
- **스케줄러 구성:**
 - APScheduler를 사용하여 정기적으로 뉴스 크롤링 및 트렌드 분석 작업 자동화.
- **크롤링 및 텍스트 처리:**
 - BeautifulSoup을 사용하여 뉴스 데이터를 크롤링.
 - konlpy를 활용하여 텍스트에서 명사 추출 및 트렌드 키워드 매칭.
- **챗봇:**

- LangSmith 를 이용한 실시간 로그 분석툴 연결
 - Langchain 을 통한 RAG 도입
-

프로젝트 마무리하며 느낀 점

- **데이터 파이프라인 구축의 어려움과 해결 과정**

뉴스 데이터를 실시간으로 크롤링하고 분석하는 과정에서 데이터의 신뢰성과 일관성을 유지하는 것이 쉽지 않았습니다. 특히, 다양한 출처에서 수집한 데이터를 통합하고 중복을 방지하며 정확한 트렌드 분석을 수행하기 위해 데이터 모델링과 정규화를 철저히 진행했습니다. 이를 통해 PostgreSQL의 효율적인 쿼리 작성법과 인덱싱 최적화 기술을 익히는 계기가 되었습니다.

- **비동기 프로그래밍과 성능 최적화**

FastAPI와 Asyncio를 활용한 비동기 처리를 통해 다량의 요청을 효율적으로 처리할 수 있도록 설계하였으며, React Query를 활용한 캐싱 및 상태 관리 전략을 도입하여 프론트엔드에서의 성능을 극대화했습니다. 이를 통해 API 응답 속도를 단축하고 사용자 경험을 개선할 수 있었습니다.

- **DevOps와 자동화 도입의 필요성**

프로젝트를 진행하면서 CI/CD 파이프라인을 구축하여 자동화된 배포 환경을 마련하였고, 이를 통해 코드의 품질을 지속적으로 유지하고 배포 프로세스를 단순화할 수 있었습니다. AWS EC2 및 Docker 기반의 배포 환경을 구성하면서 인프라 설계에 대한 이해도를 높이는 계기가 되었습니다.

- **사용자 중심의 UI/UX 설계 중요성**

React와 Tailwind CSS를 활용하여 직관적인 인터페이스를 구현하는 과정에서, 단순히 기능 구현에 초점을 맞추는 것이 아니라 사용자의 행동 흐름을 고려한 UI 설계의 중요성을 실감했습니다. 이를 위해 반복적인 사용자 피드백을 수렴하고 개선해 나갔습니다.

- **자연어 처리(NLP) 기술의 활용과 한계**

konlpy 및 OpenAI API를 활용한 트렌드 분석 및 챗봇 기능을 개발하면서 한국어 데이터에 대한 처리의 어려움을 경험했습니다. 형태소 분석의 정확성을 높이기 위해 여러 NLP 모델을 실험하며 도메인에 적합한 모델을 선택하는 과정에서 많은 인사이트를 얻을 수 있었습니다.

- **협업 및 커뮤니케이션의 중요성**

프론트엔드와 백엔드 개발을 동시에 진행하면서 여러 협업 도구(GitHub Projects, Slack, Notion 등)를 활용하여 팀원들과의 원활한 소통이 프로젝트의 성공에 중요한 요

소임을 깨달았습니다. 코드 리뷰와 문서화를 통해 프로젝트의 유지보수성과 가독성을 높이는 방법을 배웠습니다.