

Chat GPT Prompt와 Web Search 기반의 Web Editor 'Note Friends'

기획의도

(CS) 공부를 할 때 웹 검색과 Chat GPT를 통해 정보를 얻는 경우가 많습니다. 이러한 경우에 우리는 브라우저 탭 하나에는 GPT를, 또 하나에는 Web Search, 마지막에 웹 에디터를 놓고 쓰게 되어 탭을 계속해서 바꿔야하는 불편함이 있습니다. 이에 이 기능을 복합적으로 하나의 탭, 하나의 화면에 모아놓은 서비스를 만들고자 합니다.

더 이상 탭을 바꾸지 않고 하나의 화면에서 웹 검색, Chat GPT에게 질문을 진행하고 손 쉽게 공부한 내용을 저장해보세요!

또한 Chat GPT에게 질문을 직접 길게 할 필요 없이 알고싶은 키워드나 내용만 입력해도 정확하게 내가 궁금한 것들을 Chat GPT에게 물어볼 수 있도록 GPT Prompt를 사용해보세요! 만약 내가 만든 Prompt를 사용하고 싶다면 나만의 Custom Chat GPT Prompt를 통해 반복되는 질문을 키워드만 입력해도 GPT에게 정확하고 현명하게 질문할 수 있도록 해보세요!

Web Search의 경우에도 원하는 정보를 Web에서 검색한 뒤 여러 필터링을 다시 적용하여 1차 검색 결과에서 내가 원하는 검색 정보만 남길 수 있도록 빠르고 쉽게 컨트롤 해보세요!

마지막으로 Web Editor 뿐만 아니라 Design에서도 나만의 Note Template을 Custom 할 수 있도록 Sticker, BackGround, Color 등 내 개성이 들어간 Note를 Custom하고 이를 쉽게 PDF, Word 파일 등으로 변환해서 저장해보세요!

기본 기능

- 로그인
- 회원 정보 수정
- 회원 탈퇴

서비스 기능

- Note Page UI/UX
 - ChatGPT 검색 토글
 - Web Search 검색 토글
 - Editor

- ChatGPT(Vanilla)
 - GPT에게 원하는 질문을 자유롭게
 - 기존 GPT 검색을 사이트를 찾아갈 필요 없이 진행
 - (심화) GPT 대화 내용 GPT에게 기억하도록
- ChatGPT(Prompt)
 - 간단한 키워드들만 입력해도 GPT에게 정확하고 확실한 질문 전달
 - ex) Before
 - CPU 스케줄링의 개념에 대해 알려주고 이 것을 했을 때 기능이나 성능 측면에서 어떤 장점이 있는지 알려줘
 - ex) After
 - CPU 스케줄링
 - 질문을 단순화하고 질문에 대한 답변을 구조화하여 Return(표, 답변 구분)
- Web Search(Vanilla)
 - 검색 후 통상적인 결과를 Return
- Web Search(Secretary)
 - 주제와 관련된 검색 키워드 추천
 - 1차 결과를 받은 뒤 2차 필터링 기능
 - CPU 스케줄링 (검색)
 - CPU 스케줄링에 대한 개념, CPU 판매 스케줄에 맞춰 싸게 사는 법 (1차 답변)
 - 필터링 과정 ex) 꼭 포함 되어야 하는 키워드 '개념' 선택
 - CPU 판매 스케줄에 맞춰 싸게 사는 법(삭제)
 - 검색된 내용을 외부링크 없이 내부에서 보여줄 수 있는 Iframe사용하여 쉽게 스크랩 or 복사 가능하도록
- Custom Design
 - 인터랙티브한 디자인을 선택하여 Note를 꾸밀 수 있도록 함

- 사용자가 원하는 색상에 맞는 테마와 효과를 Custom으로 적용할 수 있음

심화 기능

- Note를 수정할 때 마다 자동 저장
 - DB에 특정 주기로 저장
 - Editor 내 내용이 변할 때 마다 통신
- 다른 사용자와 공유 및 토론
 - 사용자에게 Note를 공유하고 함께 보며 수정
 - Web Socket을 이용
- Quiz 생성 및 게임 Platform 제공
 - Web Socket을 이용한 퀴즈 게임 제공

기술

- Front End
 - Web Search, Chat GPT 결과 SSR
 - SSR을 통해 서버에서 응답이 완성된 뒤 보여줄 수 있도록
 - Client의 Resource를 소모하지 않도록
 - Interactive Design
 - Editor에서 Interactive한 JS을 통한 Design 적용
 - ex) 커서 움직임 혹은 스크롤 등등
 - Web Socket
 - Web Socket을 이용하여 Server에서 받은 정보를 화면으로 표현
- Back End
 - Editor 자동저장시 발생하는 Message를 Kafka를 통해 Event Broker 처리
 - Elastic Search를 통한 검색기능 강화
 - Kibana

- LogStash
- Kubernetes를 통한 Server 장애 복구(자가치유) 및 Load Balancing
 - Editor 오류 복구
- MSA를 통한 각 기능 분리
 - Chat GPT
 - Web Search
 - Editor
 - Member
 - Auth
 - Socket
- NOSQL, SQL 복합 활용
 - Editor 관련 저장
 - NOSQL 활용
 - 사용자 정보 및 기타 데이터 저장
 - SQL 활용
 - 사용자 Token
 - Redis 활용
 - 검색 기능
 - ElasticSearch