

한성대학교

컴퓨터공학부

2024- 2 SW프리캡스톤디자인

최종보고서

논문 작성 보조 서비스 Thesisfy

팀원: 김경민, 이다민, 우현호, 임혜미, 황필호

담당 교수: 김성동

1. 프로젝트 계획

SW프리캡스톤 - 1팀(팀백) 최종 프로젝트 계획서

1. 프로젝트 개요

1.1 프로젝트 이름

1.2 문제정의문

1.3 프로젝트 배경

2. 기능 구조도

2.1 기능 구조도

2.2 기능 설명

2.2.1 사용자 관리 (User Management)

2.2.2 학술 정보 제공 (Academic Content Delivery)

2.2.3 AI 시스템 (AI System)

2.2.4 보안 및 데이터 보호 (Security & Privacy)

2.2.5 알림 및 피드백 (Notifications & Feedback)

3. 시스템 구조도

3.1 시스템 구조도

3.2 시스템 구성 요소 설명

4. 일정 계획

5. 역할 분담

1. 프로젝트 개요

1.1 프로젝트 이름

- Thesisfy : 논문(Thesis) + 찾다, 발견하다(Identify)

1.2 문제정의문

- 사용자의 자연어 질문을 이해하고, 그 의도에 맞는 논문, 리포트, 기사 등을 정확하고 신속하게 찾아주는 AI 기반 학술 정보 검색 도구

1.3 프로젝트 배경

배경: 현대의 학술 연구 환경에서는 방대한 양의 논문, 리포트, 기사 등이 매일 생성되고 있다. 연구자와 학생들은 이러한 자료를 검색하고 활용하기 위해 온라인 데이터베이스와 검색 엔진에 의존한다. 그러나 기존의 키워드 기반 검색 시스템은 아래와 같은 한계가 존재한다.

- 의도 파악의 어려움: 키워드만으로는 사용자의 구체적인 질문이나 의도를 정확히 파악하기 어렵다.
- 검색결과의 과다: 관련성이 낮은 자료까지 포함되어 방대한 검색 결과가 나타나며, 원하는 정보를 찾는 데 시간이 많이 소요된다.
- 문맥 이해 부족: 동의어, 다의어 등 언어적 특성을 고려하지 않아 중요한 자료를 놓칠 수 있다.

최근 인공지능(AI)과 자연어 처리(NLP) 기술의 발전으로 이러한 문제를 해결할 수 있는 가능성이 열렸다. AI는 사용자의 자연어 질문을 이해하고, 그에 맞는 의미 기반 검색을 수행하여 더욱 정확한 결과를 제공한다.

필요성:

- 검색 효율성 향상: 사용자가 원하는 정보를 빠르게 찾을 수 있어 연구 및 학습 효율이 높아진다. 또한, 의미를 이해하는 검색으로 관련성이 높은 자료를 우선적으로 제공한다.
- 학술 연구 지원 강화: 최신 연구 동향과 관련자료를 신속하게 파악할 수 있다. 또한, 다양한 분야의 자료를 통합적으로 검색하여 혁신적인 연구를 지원한다.
- 학습자 친화적 환경 조성: 전문 용어나 복잡한 검색 기술에 익숙하지 않은 사용자도 쉽게 접근할 수 있고, 지식 수준에 관계없이 필요한 정보를 얻을 수 있어 학습 기회를 확대한다.
- 방대한 정보 관리: 증가하는 학술 자료를 효과적으로 관리하고 활용할 수 있고, 기존 연구를 정확히 파악하여 비효율적인 중복 연구를 줄인다.
- 의사결정 지원: 핵심 내용 요약과 통계 시각화를 통해 중요한 정보를 빠르게 이해할 수 있다. 사용자의 관심사에 기반한 자료를 추천하여 개인화된 연구 환경을 제공한다.

타겟 사용자:

1. 연구자 (학계 및 산업계)

- **대상:** 대학 및 연구소의 교수, 연구원, 박사과정 학생 등.
- **필요성:** 연구자들은 방대한 학술 자료에 접근하여 최신 연구 동향을 파악하고, 새로운 연구 아이디어를 탐구하기 위해 많은 시간을 검색에 할애한다. 기존의 키워드 기반 검색은 관련성이 낮은 결과를 다수 포함하기 때문에, AI 기반 학술 정보 검색 앱을 통해 의미 기반 검색을 사용하면 더욱 정확한 자료에 신속하게 접근할 수 있다. 또한, AI 기술이 자료를 요약하고 분석하여 연구자들이 필요한 핵심 정보를 빠르게 추출할 수 있다.
- **주요 요구:**
 - **효율적이고 정확한 정보 검색:** 연구자가 원하는 논문, 기사, 리포트를 신속하게 검색.

- **관련성 높은 자료 추천:** 연구 주제와 유사하거나 관련된 자료를 추천받아 연구 범위를 넓힐 수 있음.
- **빠른 최신 연구 파악:** AI가 자동으로 최신 연구 동향을 분석하여 제공함으로써 최신 정보에 대한 접근성 강화.

2. 대학생 및 대학원생

- **대상:** 학부생, 석·박사과정 대학원생 등.
- **필요성:** 학부생과 대학원생들은 학술 자료를 검색하고 분석하는 데 익숙하지 않을 수 있다. AI 기반 검색 앱은 직관적인 인터페이스와 사용자의 자연어 입력을 통해 자료 검색을 쉽게 만들어 주며, 복잡한 검색 기술을 익힐 필요 없이도 원하는 자료를 찾을 수 있도록 돕는다. 또한, 학술 논문을 요약해주거나 추천 시스템을 통해 연구와 관련된 자료를 쉽게 얻을 수 있다.
- **주요 요구:**
 - **접근성 향상:** 복잡한 검색어 사용 없이 자연어로 검색하여 원하는 자료에 쉽게 접근.
 - **효율적인 학습 지원:** 핵심 정보 요약 기능을 통해 방대한 자료를 짧은 시간 내에 학습.
 - **연관된 학습 자료 추천:** 전공 분야나 논문 주제에 맞는 관련 자료 및 참고 문헌 추천.

3. 교수 및 교육자

- **대상:** 대학 및 교육 기관의 강의 담당자, 교육 콘텐츠 개발자 등.
- **필요성:** 교수와 교육자는 학생들에게 최신 연구 자료와 논문을 제공해야 하며, 교육 자료 준비 시 신뢰할 수 있는 정보를 바탕으로 강의를 구성해야 한다. AI 검색 앱을 통해 관련성이 높은 최신 학술 정보를 손쉽게 검색할 수 있어 강의 준비 시간 단축 및 교육의 질 향상이 가능하다.
- **주요 요구:**
 - **강의 자료 준비 지원:** 강의 주제에 맞는 최신 논문, 보고서, 기사를 신속히 검색 가능.
 - **학생 맞춤형 자료 제공:** 학생 수준에 맞는 자료를 추천하고 다양한 학술 자료를 빠르게 제공.

4. 일반 학습자 및 지식 탐구자

- **대상:** 전문적인 연구자가 아닌, 특정 분야에 관심이 있는 일반 학습자 및 지식 탐구자.
- **필요성:** AI 기반 검색 앱은 비전문가들이 쉽게 접근할 수 있도록 직관적인 검색 시스템을 제공한다. 복잡한 검색어 입력 없이 자연어로 질문을 입력하면 사용자가 원하는 자료를 찾을 수 있으며, AI가 자동으로 중요한 내용을 요약하고 시각화하여 정보 습득이 용

이하다. 지식의 깊이에 상관없이 사용자는 자신에게 맞는 정보를 쉽게 찾아 학습할 수 있다.

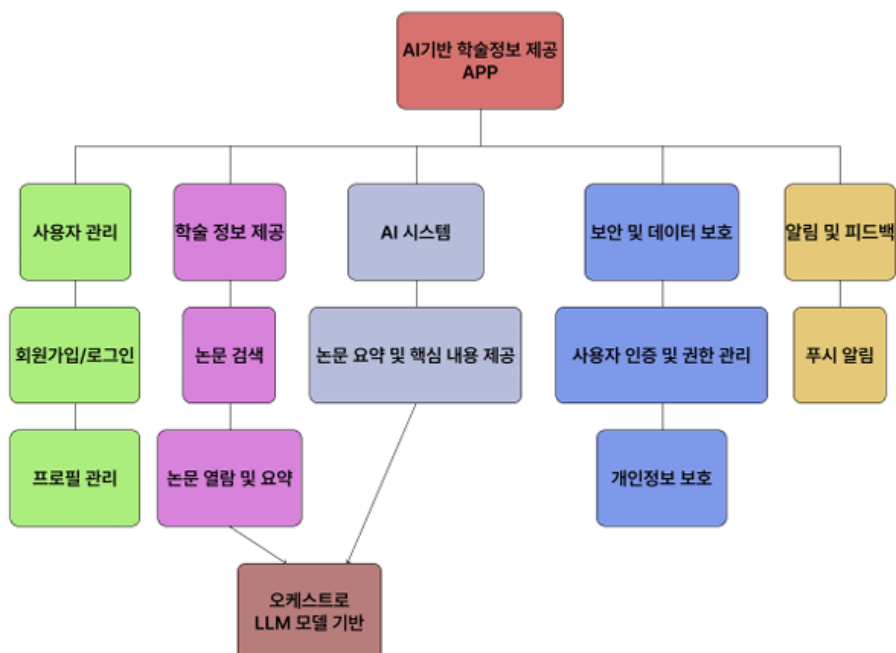
• **주요 요구:**

- **전문 용어에 대한 부담 감소:** 특정 용어나 학문적 배경이 부족해도 필요한 자료에 접근 가능.
- **효율적인 지식 탐구:** 관련된 논문이나 리포트의 요약된 정보를 빠르게 파악.
- **관심사 기반 추천:** 개인의 검색 패턴에 맞춘 맞춤형 학술 자료 추천.

결론적으로, AI 기반 학술 정보 검색 앱은 기존의 검색 방식이 가진 한계를 극복하고, 학술 정보의 접근성과 활용도를 높임으로써 연구자와 학습자에게 큰 가치를 제공한다. 이는 학술 연구의 효율성 증대와 교육 수준 향상에 기여하여 지식 사회 발전에 이바지할 것이다.

2. 기능 구조도

2.1 기능 구조도



2.2 기능 설명

2.2.1 사용자 관리 (User Management)

- 회원가입/로그인
 - 이메일을 통한 회원가입 및 로그인 기능 제공.
 - JWT 인증을 통해 사용자 인증 및 세션 관리.
 - 프로필 관리
 - 사용자 정보(이름, 이메일) 수정 및 업데이트 기능.
 - 비밀번호 변경 및 설정 관리 기능.
-

2.2.2 학술 정보 제공 (Academic Content Delivery)

- 논문 검색
 - 기본적인 키워드 기반 논문 검색 기능 제공.
 - 검색 정렬은 최신순 제공.
 - 논문 열람 및 요약
 - 논문 내용을 보기 쉽게 요약하여 제공.
-

2.2.3 AI 시스템 (AI System)

- 논문 요약 및 핵심 내용 제공
 - LLM 기반 AI 기능을 활용하여 논문 요약 생성.
-

2.2.4 보안 및 데이터 보호 (Security & Privacy)

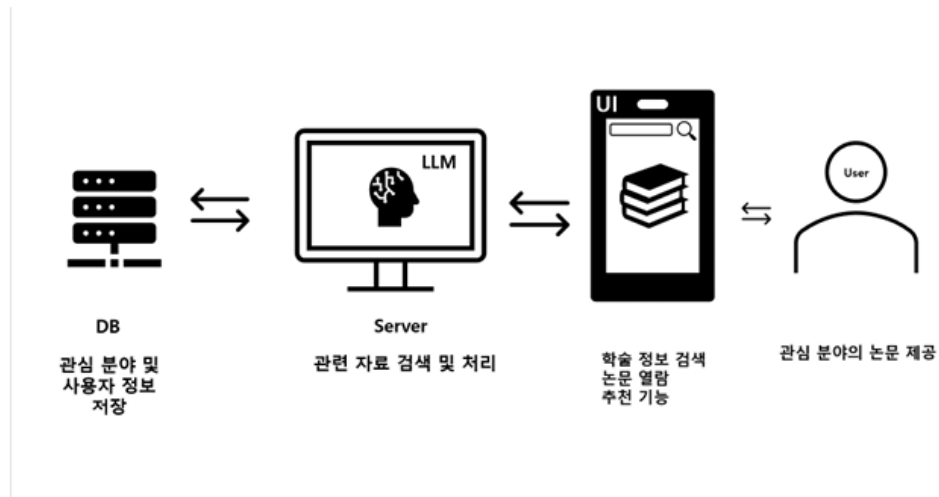
- 사용자 인증 및 권한 관리
 - JWT를 활용한 인증 시스템으로 안전한 사용자 인증 처리.
 - 개인정보 보호
 - GDPR 및 관련 법규에 따른 개인정보 보호 정책 준수.
-

2.2.5 알림 및 피드백 (Notifications & Feedback)

- 푸시 알림
 - 새로운 논문 추가, 관심 논문 업데이트 알림 제공.

3. 시스템 구조도

3.1 시스템 구조도



3.2 시스템 구성 요소 설명

| 구성 요소 | 설명 | 개발 환경 및 구현 방법 |
|----------------|--|--|
| 서버 | AI 기능을 처리하는 서버로, 사용자 요청을 받아 관련 자료를 검색하고 처리하는 역할을 수행한다. | Node.js를 사용하여 IntelliJ환경에서 개발한다. 사용자에게 입력받은 정보를 바탕으로 동적인 콘텐츠를 생성하고, 데이터베이스와 상호 작용한다. |
| 데이터베이스 | 사용자 정보, 논문 데이터, 추천 시스템 데이터를 저장하고 관리하는 역할을 한다. | MySQL을 사용하여 관계형 데이터베이스를 구축한다. Sequelize 등의 ORM을 통해 Node.js 서버와 데이터베이스를 연동하고, 데이터 모델링과 스키마 설계를 수행한다. |
| 사용자 인터페이스 (UI) | 사용자에게 학술 정보 검색, 논문 열람, 추천 등을 제공하는 프론트엔드 애플리케이션이다. | Swift를 사용하여 Xcode에서 네이티브 iOS 앱을 개발한다. UIKit 또는 SwiftUI로 UI를 구현하고, Alamofire 등 네트워크 라이브러리를 활용하여 서버와 RESTful API 통신을 한다. |
| AI 및 LLM | 논문 요약, 핵심 내용 제공, 추천 시스템 등 AI 기능을 구현하기 | 오픈스트로 기업에서 LLM 서비스를 서버에서 호출하여 AI 관련 처리를 수행한 |

| 구성 요소 | 설명 | 개발 환경 및 구현 방법 |
|-----------------|---|--|
| | 위한 서비스이다. | 다. 비동기 프로그래밍과 캐싱 전략을 적용하여 성능을 최적화하고, 처리 결과를 서버를 통해 iOS 앱에 전달한다. |
| 외부 API 및 서비스 | 논문 데이터 수집 및 추가 정보를 제공하기 위한 외부 API와 서비스이다. | Crossref, Semantic Scholar API 등을 서버에서 호출하여 논문 메타데이터와 관련 정보를 수집한다. 필요한 경우 OAuth 등의 인증 방식을 사용하여 안전하게 통합한다. |
| 보안 및 인증 시스템 | 사용자 인증, 권한 부여, 데이터 보안을 담당하는 시스템이다. | JWT(Json Web Token)를 사용하여 인증 및 세션 관리를 구현한다. HTTPS를 적용하여 데이터 전송 보안을 강화하고, 입력 검증을 통해 보안 취약점을 최소화한다. |
| 프로젝트 관리 및 협업 도구 | 개발 팀의 효율적인 협업과 프로젝트 관리를 지원한다. | Git과 GitHub를 사용하여 버전 관리를 하고, Jira나 Trello로 작업 현황을 추적한다. Discord나 Google Meet를 통해 팀원 간의 커뮤니케이션을 원활하게 한다. |
| 테스트 및 배포 환경 | 애플리케이션의 품질 보증과 안정적인 배포를 지원한다. | Jest, Mocha 등의 테스트 프레임워크를 사용하여 유닛 테스트와 통합 테스트를 진행한다. Jenkins나 GitHub Actions를 통해 CI/CD 파이프라인을 구축하고, AWS나 Heroku 등 클라우드 서비스를 활용하여 서버와 데이터베이스를 배포한다. |
| 추천 시스템 | 사용자 회원가입 시 관심있는 토픽을 지정하여 관련 학술 자료를 추천한다. | DB에 토픽 데이터를 넣어놓고, 회원가입을 할 때, 사용자마다 지정한 토픽을 토대로 해당 토픽이 포함된 학술 자료를 추천. |

4. 일정 계획

| 주차 | 마일스톤/작업 | 설명 |
|-----|-------------------|--|
| 1주차 | 프로젝트 계획 수립 및 팀 구성 | 프로젝트의 목표와 비전을 명확히 설정하고 팀 구성을 완료한다. 필요한 개발자, 디자이너, DB 담당자 등을 모집하여 각자의 역할과 책임을 분담하며, 효 |

| | | |
|----------|-----------------|--|
| | | 올적인 협업을 위한 프로젝트 관리 도구와 커뮤니케이션 채널을 선정한다. |
| 2주차~4주차 | 요구사항 분석 및 | 시장 조사와 요구사항 분석을 진행한다. 유사한 앱을 분석하여 차별화 포인트를 찾고, 잠재 사용자들과의 인터뷰나 설문조사를 통해 필요한 기능과 기대 사항을 파악한다. 이를 바탕으로 기능 목록을 작성하고 우선순위를 결정한다. |
| 5주차 | 프로젝트 계획서 작성 | 세부적인 프로젝트 계획서를 작성한다. 각 기능 개발에 필요한 시간과 인력을 배분하여 일정과 마일스톤을 설정하고, 발생할 수 있는 위험 요소를 식별하여 대응 전략을 마련한다. 또한 필요한 자원과 비용을 산출하여 최종 계획을 수립한다. |
| 6주차 | 시스템 아키텍처 및 db설계 | 시스템 아키텍처와 데이터베이스 설계를 진행한다. 앱의 전체 구조와 데이터 흐름을 설계하여 모듈 간 연계를 명확히 하고, 사용자 정보, 논문 데이터, 추천 알고리즘 등을 위한 데이터베이스 스키마를 작성한다. 필요한 API와 외부 서비스의 통합 방안도 마련한다. |
| 7주차~10주차 | 주요 모듈 개발 | 주요 모듈 개발에 집중합니다. 검색 기능, 필터링 및 정렬 옵션, LLM을 활용한 논문 요약 및 핵심 내용 제공, 추천 시스템, 비슷한 주제 분류 기능 등을 개발한다. 또한 필요시 다국어 지원 및 |

| | | |
|-----------|-----------------|---|
| | | 번역 기능, 인용 및 참고문헌 관리, 보안 및 개인정보 보호 기능도 구현한다. |
| 11주차~13주차 | 통합 테스트 및 피드백 개선 | 통합 테스트와 피드백 개선을 진행한다. 개발된 모듈들을 통합하여 시스템 전체의 기능과 성능을 테스트하고, 베타 버전을 통해 실제 사용자들의 피드백을 수집한다. 이를 토대로 발견된 이슈를 수정하고 앱의 기능성과 사용자 경험을 향상시킨다. |
| 14주차 | 최종 릴리스 및 배포 | 최종 릴리스와 배포를 진행한다. 결과물 발표 및 평가를 위한 앱의 모든 기능이 의도대로 작동하는지 최종 검수를 완료한다. |

5. 역할 분담

1. 김경민 - 프론트 iOS 앱 개발
2. 우현호 - 백엔드 개발, db관리
3. 이다민 - 백엔드 개발, 서버관리
4. 임혜미 - 백엔드 개발, UI/UX개발, 기획, 앱 디자인
5. 황필호 - 프론트 iOS 앱 개발



SW 상세설계서 - 1팀 (팀백)

목차

목차

1. 서론

- 1.1 목적 (Purpose)
- 1.2 범위 (Scope)
- 1.3 참고 문서 (Reference Documents)

2. 시스템 개요

- 2.1 기능 설명 (Functional Overview)
- 2.2 시스템 아키텍처 (System Architecture)

3. 모듈 설계

- 3.1 사용자 관리 모듈 (User Management Module)
- 3.2 챗봇 논문 요약 및 검색 모듈 (Chatbot Summary and Search Module)
- 3.3 논문 검색 및 열람 모듈 (Paper Search and Access Module)

4. 데이터베이스 설계

- 4.1 사용자 테이블 (User Table)
- 4.2 논문 테이블 (Content Table)
- 4.3 Chat Log 테이블 (Chat Log table)
- 4.4 App Log 테이블 (App Log table)

5. 시퀀스 다이어그램

- 5.1 채팅형 논문 검색 시퀀스
- 5.2 논문 키워드 검색 시퀀스

6. API 설계

- 6.1 사용자 관리 API
- 6.2 학술 정보 제공 API
- 6.3 AI 시스템 API
- 6.4 보안 및 데이터 보호 API
- 6.5 알림 및 피드백 API

7. 보안 설계

8. 성능 및 확장성 고려사항

- 8.1 성능 고려사항
- 8.2 확장성 고려사항
- 8.3 추가 고려사항

9. 테스트 계획

10. 배포 계획

1. 서론

1.1 목적 (Purpose)

이 설계서의 목적은 **Thesisfy** 시스템의 각 구성 요소를 구체적으로 설계하여 개발자들이 소프트웨어를 효율적이고 일관되게 구현할 수 있도록 돕는 것입니다. 이를 통해 시스템의 주요 기능, 데이터 흐름, 인터페이스, 보안 요구사항을 명확히 정의합니다.

1.2 범위 (Scope)

이 설계서는 **Thesisfy**의 백엔드 및 iOS 프론트엔드 모듈 설계를 포함하며, 시스템의 기능적 요구사항과 비기능적 요구사항을 모두 다룹니다. 또한, 데이터베이스 구조, 보안 설계, 성능 최적화, 테스트 계획 등을 포함하여 전반적인 시스템 개발을 위한 가이드를 제공합니다.

1.3 참고 문서 (Reference Documents)

- **Thesisfy** 프로젝트 계획서
- 소프트웨어 요구사항 명세서 (SRS)
- 시스템 아키텍처 설계 문서
- Node.js, MySQL 및 Swift 관련 기술 문서
- GDPR 및 관련 보안 지침

2. 시스템 개요

2.1 기능 설명 (Functional Overview)

Thesisfy는 사용자의 자연어 질문을 이해하고, 그 질문에 맞는 논문, 리포트, 기사 등을 정확하고 신속하게 찾아주는 AI 기반 학술 정보 검색 도구입니다. 주요 기능은 다음과 같습니다:

- **사용자 관리:**
 - 로그인 및 회원가입
 - 사용자 프로필 관리
- **학술 정보 제공:**
 - AI 기반 논문 검색 및 필터링
 - 논문 열람 및 자동 요약
- **AI 시스템:**
 - LLM을 활용한 논문 요약 및 핵심 내용 제공
- **보안 및 데이터 보호:**
 - JWT 인증을 통한 사용자 인증 및 권한 관리
 - GDPR 준수를 통한 개인정보 보호
- **알림 및 피드백:**
 - 푸시 알림을 통해 새로운 논문 업데이트 및 관심 논문 알림 제공

2.2 시스템 아키텍처 (System Architecture)

시스템은 클라이언트-서버 아키텍처로 구성되며, 다음과 같은 주요 컴포넌트로 나뉩니다:

- **사용자 인터페이스 (UI):** iOS 애플리케이션의 사용자 인터페이스로, 사용자가 논문 검색 및 요약을 요청하고 결과를 확인할 수 있는 인터페이스를 제공합니다.
- **서버:** Node.js를 기반으로 사용자의 요청을 처리하고, AI 시스템 및 데이터베이스와의 통합을 통해 비즈니스 로직을 구현합니다.
- **데이터베이스:** MySQL을 사용하여 사용자 정보, 논문 데이터, 검색 히스토리, 추천 데이터 등을 관리하며 CRUD 작업을 수행합니다.
- **AI 서버:** 오픈스트로에서 제공한 LLM API를 활용해 텍스트 분석 및 응답을 생성하여 사용자의 질문에 맞는 논문 요약 및 핵심 내용을 제공합니다.

3. 모듈 설계

3.1 사용자 관리 모듈 (User Management Module)

- **기능 설명:** 회원가입, 로그인, 프로필 관리 기능을 제공합니다.
- **입력 (Input):** 사용자 정보 (이메일, 비밀번호, 이름 등)
- **출력 (Output):** 인증 토큰, 프로필 정보
- **프로세스 흐름:**
 1. 사용자가 회원가입 정보를 제출하면, 서버는 이를 데이터베이스에 저장합니다.
 2. 비밀번호는 bcrypt 해시 알고리즘을 통해 암호화됩니다.
 3. 로그인 시, 사용자가 입력한 비밀번호와 해시된 비밀번호를 비교하여 인증합니다.
- **주요 클래스 및 메서드:**
 - **UserController:**
 - `registerUser(UserDto user)`: 회원가입 기능
 - `login(String email, String password)`: 로그인 기능
 - **UserService:**

- `validateUserCredentials()`: 사용자 자격 검증
- `createUser()`: 사용자 생성
- **UserRepository**:
 - `save(UserEntity user)`: 사용자 정보 저장
 - `findByEmail(String email)`: 이메일로 사용자 검색

3.2 챗봇 논문 요약 및 검색 모듈 (Chatbot Summary and Search Module)

- **기능 설명**: 사용자의 자연어 질문을 이해하고, AI 챗봇이 해당 질문에 맞는 논문을 요약해 주고 관련 자료를 찾아줍니다.
- **입력 (Input)**: 사용자 질문 (자연어 형태)
- **출력 (Output)**: 논문 요약, 논문 목록 (제목, 저자, 요약, 논문 링크 등)
- **프로세스 흐름**:
 1. 사용자가 자연어로 질문을 입력합니다.
 2. AI 챗봇이 NLP(자연어 처리) 모델을 통해 질문의 의도를 파악합니다.
 3. 질문에 맞는 논문을 검색하고, 논문의 핵심 내용을 요약하여 사용자에게 보여줍니다.
 4. 추가로 관련 논문 목록을 제공하여 사용자가 원하는 자료를 선택할 수 있도록 합니다.
- **주요 클래스 및 메서드**:
 - **ChatbotController**
 - `processQuery(String query)`: 사용자 질문을 처리하여 결과 반환.
 - **ChatbotService**
 - `analyzeIntent(String query)`: 사용자의 질문을 분석하여 검색 의도 파악.
 - `fetchAndSummarizeContent(String query)`: 검색된 논문에 대해 요약 생성.
 - **ChatbotRepository**
 - `findRelevantPapers(String keywords)`: 키워드 기반으로 논문 검색.
 - `getSummaryById(int paperId)`: 논문 ID에 따라 요약된 정보 반환.

3.3 논문 검색 및 열람 모듈 (Paper Search and Access Module)

- **기능 설명**: 검색된 논문에 대한 전체 내용을 열람할 수 있는 URL을 제공하는 모듈입니다.
- **입력 (Input)**: 논문 ID 또는 논문 선택 (사용자가 선택한 논문)
- **출력 (Output)**: 전체 논문 열람 URL
- **프로세스 흐름**:
 1. 사용자가 논문 목록에서 원하는 논문을 선택합니다.
 2. 선택한 논문에 대한 전체 URL을 조회하여 제공합니다.
 3. 사용자는 제공된 링크를 통해 논문을 열람할 수 있습니다.
- **주요 클래스 및 메서드**:
 - **SearchController**
 - `viewPaperDetails(int paperId)`: 논문 상세 정보를 반환.
 - `getFullPaperUrl(int paperId)`: 논문의 전체 열람 URL 제공.
 - **SearchService**
 - `getPaperById(int paperId)`: 선택된 논문에 대한 상세 정보 조회.
 - **SearchRepository**
 - `fetchPaperDetails(int paperId)`: 논문 ID로 논문 데이터 조회.
 - `fetchPaperUrl(int paperId)`: 논문 ID로 전체 논문 URL 조회.

4. 데이터베이스 설계



```
-- 사용자 테이블 (User Table)
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY COMMENT '사용자 고유 ID (자동 증가)',
  email VARCHAR(255) NOT NULL UNIQUE COMMENT '사용자 이메일 (유니크)',
  password VARCHAR(255) NOT NULL COMMENT '암호화된 비밀번호',
  name VARCHAR(100) NOT NULL COMMENT '사용자 이름',
  created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '계정 생성 시간'
);

-- 논문 테이블 (Content Table)
CREATE TABLE contents (
  id INT AUTO_INCREMENT PRIMARY KEY COMMENT '논문 고유 ID (자동 증가)',
  title VARCHAR(255) NOT NULL COMMENT '논문 제목',
  summary TEXT COMMENT '논문 요약',
  author VARCHAR(100) NOT NULL COMMENT '논문 저자',
  published_date DATETIME NOT NULL COMMENT '논문 출판 날짜'
);

-- 채팅 로그 테이블 (Chat Log Table)
CREATE TABLE chat_logs (
  room_name VARCHAR(255) PRIMARY KEY COMMENT '채팅방 고유코드 (랜덤 생성)',
  chat TEXT NOT NULL COMMENT '대화 내용 작성',
  date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '채팅 시작한 날짜 저장',
  keyword VARCHAR(255) COMMENT 'LLM이 생성한 토픽이나 키워드 저장',
  user_id INT NOT NULL COMMENT '사용자 ID',
  FOREIGN KEY (user_id) REFERENCES users(id)
);

-- 앱 로그 테이블 (App Log Table)
CREATE TABLE app_logs (
  id INT AUTO_INCREMENT PRIMARY KEY COMMENT '로그 고유 ID (자동 증가)',
  unsolved_problem TEXT COMMENT 'LLM이 대답하지 못한 질문들 저장',
  feedback TEXT COMMENT '사용자 피드백',
  date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '앱 출입 시간 기록',
  user_id INT NOT NULL COMMENT '사용자 ID',
  FOREIGN KEY (user_id) REFERENCES users(id)
);
```

4.1 사용자 테이블 (User Table)

| 컬럼명 | 데이터 타입 | 길이 | 기본값 | PK | FK | not NULL | 설명 |
|----------|---------|-----|-----|----|----|----------|----------------|
| ID | INT | - | - | Y | N | A | 사용자 ID (자동 증가) |
| Email | VARCHAR | 255 | - | N | N | A | 사용자 이메일 |
| Password | VARCHAR | 255 | - | N | N | A | 암호화된 비밀번호 |
| Name | VARCHAR | 100 | - | N | N | A | 사용자 이름 |

| | | | | | | | |
|-----|---------|-----|---|---|---|---|-------|
| JOB | VARCHAR | 100 | - | N | N | A | 사용자 직 |
|-----|---------|-----|---|---|---|---|-------|

4.2 논문 테이블 (Content Table)

| 컬럼명 | 데이터 타입 | 길이 | 기본값 | PK | FK | not NULL | 설명 |
|----------------|----------|-----|-----|----|----|----------|----------------|
| ID | INT | - | - | Y | N | A | 논문 고유 동 증가) |
| Title | VARCHAR | 255 | - | N | N | A | 논문 제목 |
| Summary | TEXT | - | - | N | N | D | 논문 요약 |
| Author | VARCHAR | 100 | - | N | N | A | 논문 저자 |
| Published_date | DATETIME | - | - | N | N | A | 논문 출판 |

4.3 Chat Log 테이블 (Chat Log table)

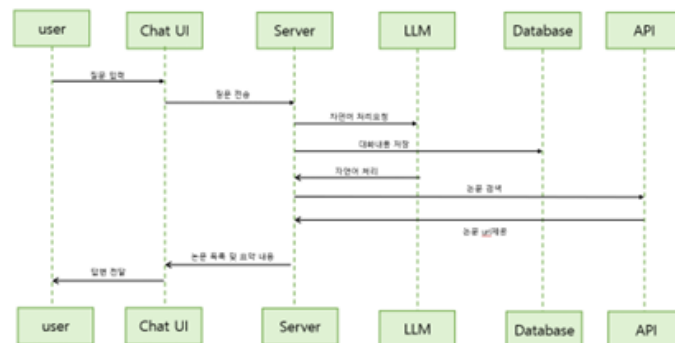
| 컬럼명 | 데이터 타입 | 길이 | 기본값 | PK | FK | not NULL | 설명 |
|-----------|---------|----|-----|----|----|----------|----------------------|
| Room name | varchar | - | - | Y | N | A | 채팅방 고 (랜덤 생성 |
| Chat | varchar | - | - | N | N | D | 대화 내용 |
| Date | varchar | - | - | N | N | A | 채팅 시작 저장 |
| Keyword | varchar | - | - | N | N | D | LLM이 생 픽이나 키 장 |
| Nickname | varchar | - | - | N | N | A | 사용자 닉 |
| (빈 컬럼) | varchar | n | - | N | D | - | (빈 컬럼이 설명 작성 |

4.4 App Log 테이블 (App Log table)

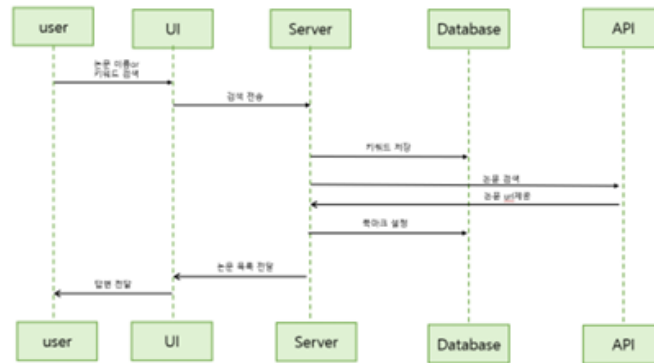
| 컬럼명 | 데이터 타입 | 길이 | 기본값 | PK | FK | not NULL | 설명 |
|------------------|---------|----|-----|----|----|----------|---------------|
| Unsolved_problem | varchar | - | - | N | N | A | LLM이 못한 질문 |
| Feedback | varchar | - | - | N | N | D | 사용자 |
| Date | varchar | - | - | N | N | A | 앱 종료 |

5. 시퀀스 다이어그램

5.1 채팅형 논문 검색 시퀀스



5.2 논문 키워드 검색 시퀀스



6. API 설계

6.1 사용자 관리 API

• 회원가입 API

◦ URL: `/api/users/register`

◦ Method: `POST`

◦ Request Body:

```
// json
{
  "email": "user@example.com",
  "password": "password123",
  "name": "John Doe"
}
```

◦ Response:

```
// json
{
  "message": "User registered successfully",
  "userId": 1
}
```

◦ 에러 처리:

- 이메일 중복 시 400 에러 반환
- 유효성 검사 실패 시 422 에러 반환

• 로그인 API

◦ URL: `/api/users/login`

◦ Method: `POST`

◦ Request Body:


```
// json
{
  "email": "user@example.com",
  "password": "password123"
}
```

◦ **Response:**

```
// json
{
  "message": "Login successful",
  "token": "jwt-token-here"
}
```

◦ **에러 처리:**

- 이메일 또는 비밀번호 불일치 시 401 에러 반환

• **프로필 조회 API**

◦ **URL:** `/api/users/profile`

◦ **Method:** `GET`

◦ **Headers:** `Authorization: Bearer {token}`

◦ **Response:**

```
// json
{
  "userId": 1,
  "email": "user@example.com",
  "name": "John Doe",
  "nickname": "Scholar123",
  "job": "Graduate Student",
  "affiliation": "XYZ University"
}
```

6.2 학술 정보 제공 API

• **논문 검색 API**

◦ **URL:** `/api/contents/search`

◦ **Method:** `POST`

◦ **Request Body:**

```
// json
{
  "query": "Impact of AI in education",
  "filters": {
    "type": "thesis",
    "date_range": {
      "from": "2020-01-01",
      "to": "2024-12-31"
    }
  }
}
```

```
}
```

◦ **Response:**

```
// json
{
  "message": "Search completed",
  "results": [
    {
      "id": 101,
      "title": "AI's Role in Modern Education",
      "summary": "This paper explores...",
      "author": "Jane Doe",
      "published_date": "2023-06-15",
      "source_url": "https://example.com/thesis123"
    },
    ...
  ]
}
```

• **논문 열람 및 요약 API**

◦ **URL:** `/api/contents/{id}`

◦ **Method:** `GET`

◦ **Response:**

```
// json
{
  "id": 101,
  "title": "AI's Role in Modern Education",
  "full_text": "...",
  "summary": "This paper explores..."
}
```

6.3 AI 시스템 API

• **논문 요약 API**

◦ **URL:** `/api/ai/summary`

◦ **Method:** `POST`

◦ **Request Body:**

```
// json
{
  "contentId": 101
}
```

◦ **Response:**

```
// json
{
  "summary": "This paper explores..."
}
```

```
}
```

6.4 보안 및 데이터 보호 API

• JWT 인증 API

- URL: `/api/auth/verify`
- Method: `POST`
- Request Body:

```
// json
{
  "token": "jwt-token-here"
}
```

- Response:

```
// json
{
  "message": "Token is valid"
}
```

6.5 알림 및 피드백 API

• 논문 알림 API

- URL: `/api/notifications/subscribe`
- Method: `POST`
- Request Body:

```
// json
{
  "userId": 1,
  "contentId": 101
}
```

- Response:

```
// json
{
  "message": "Subscribed to content notifications successfully"
}
```

• 피드백 제출 API

- URL: `/api/feedback`
- Method: `POST`
- Request Body:

```
// json
{
  "userId": 1,
```

8.1 성능 고려사항

- **빠른 응답 속도:** 사용자 요청에 대한 신속한 응답을 위해 챗봇의 검색 알고리즘을 최적화하고, 캐싱 기술을 사용하여 자주 검색되는 논문 데이터를 빠르게 제공해야 합니다.
- **데이터베이스 최적화:** 대량의 논문 데이터를 저장하고 검색할 수 있도록 데이터베이스 설계를 효율적으로 하고, 인덱싱을 통해 검색 속도를 향상시킵니다. 관계형 데이터베이스의 경우, 적절한 인덱스 및 쿼리 최적화가 필수입니다.
- **부하 분산:** 트래픽이 많아질 경우를 대비해, 로드 밸런싱을 통해 여러 서버로 요청을 분산시켜 성능을 유지합니다. AWS, GCP 등 클라우드 서비스의 로드 밸런싱 서비스를 활용할 수 있습니다.
- **캐싱:** Redis나 Memcached와 같은 인메모리 캐싱 시스템을 사용해 반복적으로 요청되는 데이터에 대해 빠른 접근을 제공합니다.
- **API 속도 최적화:** 외부 API와의 통신 속도를 최적화하기 위해 비동기 처리나 적절한 요청 큐를 사용하여 API의 대기 시간을 줄입니다.

8.2 확장성 고려사항

- **모듈화된 아키텍처:** 챗봇의 각 기능을 독립적인 모듈로 설계하여 필요에 따라 기능을 추가하거나 교체할 수 있게 합니다. 예를 들어, 검색 알고리즘, 데이터 수집 모듈, 사용자 인터페이스를 분리하여 관리할 수 있도록 합니다.
- **수평적 확장성:** 서버나 데이터베이스를 쉽게 확장할 수 있도록 오케스트로 클라우드 기반 인프라를 이용할 예정입니다.
- **마이크로서비스 아키텍처:** 챗봇의 각 기능을 독립적인 마이크로서비스로 개발하여 서로 독립적으로 확장할 수 있게 합니다. 예를 들어, 사용자 관리, 검색 기능, 데이터 수집 기능 등을 각기 다른 서비스로 분리할 수 있습니다.
- **데이터베이스 샤딩:** 대용량 데이터를 처리할 수 있도록 샤딩을 사용하여 데이터베이스를 여러 서버에 분산합니다. 특히 데이터 양이 방대할 경우, 논문 ID 등을 기준으로 데이터를 나누어 각 서버에서 처리하도록 설계합니다.

8.3 추가 고려사항

- **API 요청 제한 관리:** 외부 API를 사용하는 경우, API 요청 제한(쿼터)을 고려하여 트래픽 제어를 위해 요청 수를 관리할 수 있는 로직을 구현합니다.
- **사용자 데이터 및 보안 관리:** 사용자 데이터를 안전하게 처리하고 저장하기 위해 SSL 암호화를 통해 보안을 강화하고, 필요에 따라 개인 정보 보호 규정에 따라 데이터를 관리합니다.
- **모니터링 및 알림 시스템:** 시스템 성능과 오류를 실시간으로 모니터링하고, 장애가 발생했을 때 알림을 받을 수 있도록 시스템 상태를 체크할 수 있는 모니터링 툴을 설치합니다. 예를 들어, Prometheus와 Grafana를 활용한 모니터링을 구현할 수 있습니다.

9. 테스트 계획

- **유닛 테스트:**
 - 주요 서비스 클래스의 개별 기능에 대한 유닛 테스트를 수행합니다.
 - 예: 사용자 생성 기능의 유효성 검사, 논문 검색 알고리즘의 정확성 테스트.
- **통합 테스트:**
 - 모듈 간 연동 및 API 통합 테스트를 진행하여 시스템의 연계성을 검증합니다.
 - 예: 사용자 로그인 후 논문 검색 및 요약 기능이 올바르게 작동하는지 확인.
- **테스트 도구:**
 - JUnit, Mockito를 사용하여 백엔드 유닛 및 통합 테스트 수행.
 - Postman을 통해 API 호출 및 통합 테스트 진행.

10. 배포 계획

- **배포 환경:**
 - AWS EC2 인스턴스를 사용하여 서버와 데이터베이스를 배포합니다. Node.js 백엔드 서버와 MySQL 데이터베이스가 포함된 환경을 설정하여 안정적인 학술 정보 검색 시스템을 운영합니다.
- **배포 자동화:**

```
    "contentId": 101,
    "feedback": "This article was very insightful!"
}
```

◦ Response:

```
// json
{
    "message": "Feedback submitted successfully"
}
```


7. 보안 설계

| 보안 항목 | 설명 | 보안 대책 | 구현 단계 |
|---------------------|--|---|---|
| 1. 사용자 인증 및 권한 관리 | JWT(JSON Web Token)를 사용하여 안전하게 사용자 인증과 세션 관리를 수행합니다. 사용자 역할(관리자, 일반 사용자)에 따라 권한을 구분하여 관리합니다. | - JWT를 사용하여 로그인 시 토큰을 발급하고 API 요청 시 인증 및 권한을 검증- 역할 기반 접근 제어(Role-Based Access Control, RBAC)를 통해 사용자의 권한에 따라 기능 접근을 제한 | 1. 로그인/회원가입 API에 JWT 발급 로직 추가2. API 호출 시 JWT 인증 필터 추가3. RBAC 로직으로 사용자 역할 구분 |
| 2. 비밀번호 암호화 | 사용자의 비밀번호는 안전하게 암호화하여 데이터베이스에 저장하고 인증 시 비교합니다. | - 비밀번호는 bcrypt 알고리즘을 사용하여 해시 처리 후 저장- 인증 시 사용자가 입력한 비밀번호와 해시된 비밀번호를 비교하여 검증 | 1. bcrypt 라이브러리를 사용하여 비밀번호 해시 처리2. 로그인 API에서 비밀번호 비교 로직 구현 |
| 3. 데이터 전송 보안 | HTTPS를 사용하여 클라이언트와 서버 간의 모든 통신을 암호화합니다. | - SSL/TLS 인증서를 적용하여 모든 API와 클라이언트-서버 간 통신을 HTTPS로 보호- 중요한 정보가 전송될 때는 추가적인 암호화 알고리즘(AES)을 사용하여 이중 보호 | 1. 서버에 SSL 인증서 적용 및 HTTPS 설정2. 데이터 전송 시 AES 암호화 모듈 적용 |
| 4. 입력값 검증 및 보호 | SQL 인젝션, XSS 등 공격을 방지하기 위해 모든 입력값을 검증하고 필터링합니다. | - 모든 사용자 입력값에 대해 서버 및 클라이언트 측 검증 로직 구현- SQL 인젝션 방지를 위해 파라미터화된 쿼리 사용 및 ORM 활용- HTML, JS 응답 시 XSS 필터링 및 인코딩 처리 | 1. 클라이언트 측에서 기본적인 유효성 검사 구현2. 서버에서 파라미터화된 쿼리와 ORM을 통한 SQL 인젝션 방어3. XSS 방지 필터 적용 |
| 5. 데이터 저장 보안 | 사용자 정보, 검색 기록 등 민감한 데이터는 안전하게 저장됩니다. | - 사용자의 민감한 정보는 데이터베이스에 AES 암호화를 사용하여 저장- 검색 기록은 암호화된 상태로 관리하여 사용자 프라이버시 보호 | 1. AES 암호화 알고리즘을 적용하여 민감한 데이터 암호화 저장2. 검색 기록 저장 로직에 암호화 적용 |
| 6. 파일 업로드 보안 | 사용자 프로필 이미지와 같은 파일 업로드 기능에서 악성 파일 업로드를 방지합니다. | - 업로드된 파일의 확장자 및 파일 크기 제한- 실행 가능한 파일 업로드 차단- 업로드된 파일은 임의의 파일명으로 저장하고, 외부에서 접근할 수 없는 경로에 저장 | 1. 파일 업로드 시 확장자와 크기 제한 로직 구현2. 저장 경로 설정 및 랜덤 파일명 생성 로직 구현 |
| 7. 웹 서비스 요청 및 결과 검증 | API 요청 및 응답에서 사용자 데이터를 안전하게 처리합니다. | - CSRF(Cross-Site Request Forgery) 방지를 위해 CSRF 토큰 사용- 모든 API 요청에 대해 요청 검증 로직을 포함하고, 잘못된 요청에 대해서는 오류 메시지를 반환 | 1. API 요청 시 CSRF 토큰 검증 로직 추가2. 잘못된 요청에 대해 400/403 에러 코드 반환 |
| 8. 중요 정보 전송 및 보호 | 중요 정보를 네트워크로 전송할 때 암호화하여 안전하게 보호합니다. | - 세션 관리에 사용되는 토큰은 반드시 HTTPS를 통해 전송- 중요한 데이터(예: 비밀번호 변경 요청)는 AES 암호화 후 전송 | 1. HTTPS 적용 후 토큰 전송 구현2. 비밀번호 변경 및 민감한 데이터 전송 시 추가적인 AES 암호화 적용 |
| 9. 권한 검증 및 중요 기능 보호 | 비밀번호 변경, 검색 등 중요 기능은 적절한 권한 검증을 통해 보호됩니다. | - 중요 기능 호출 시마다 사용자 권한 및 유효성을 검증- CSRF 방지를 위해 중요 기능 수행 시 추가 검증 토큰 요구 | 1. 중요 기능 API에 권한 검증 로직 포함2. CSRF 토큰 기반 추가 검증 로직 적용 |
| 10. 모니터링 및 로그 관리 | 시스템 내 발생하는 보안 이벤트와 오류를 기록하고 모니터링합니다. | - 인증 실패, 비정상적인 API 접근, 데이터 변조 시도 등의 보안 이벤트를 로그로 기록- 실시간 모니터링 시스템(New Relic 등)을 통해 서버 상태를 감시 | 1. 시스템 로그 파일 저장 및 보안 이벤트 로깅 구현2. 모니터링 시스템 통합 및 실시간 알림 설정 |

8. 성능 및 확장성 고려사항

- CI/CD 파이프라인을 구축하여 GitHub Actions 또는 Jenkins를 사용해 자동 배포를 설정합니다.
- 코드 변경 시, 자동으로 빌드, 테스트합니다.
- Docker를 사용하여 애플리케이션을 컨테이너화하고, AWS ECS와 같은 컨테이너 오케스트레이션 서비스를 통해 확장성을 보장합니다.

3. 최종 발표 자료 (리허설 용 자료, 없으면 타 팀의 자료 참고하여 작성)




논문 작성 보조 서비스 Thesisfy

팀원 김경민 이다인 우원호 임혜미 황필호


목차


- 프로젝트 개요
- 주요 기능
- 시장 분석
- SWOT 분석
- 향후 계획



프로젝트 개요


01

 프로젝트 개요




의도 파악의 어려움

키워드만으로 사용자의 구체적 질문이나 의도 파악이 어렵다




검색 결과의 과다

방대한 검색 결과 때문에 원하는 정보만 찾기가 어렵다




문맥 이해 부족

언어적 특성을 고려하지 않아 중요한 자료를 놓칠 수 있다

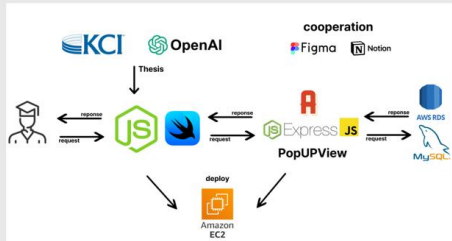



주요 기능

02

 주요 기능


기술 스택




 주요 기능

메인화면

이메일을 통한 회원 가입 및 로그인
사용자 정보 수집에 관한 약관 등의 사항 제시
3단계로 구성된 사용 가이드 제공



 주요 기능

챗봇 / 사이드바 / 검색

질문 시 AI가 질문이나 요청에 응답
논문 관련되지 않은 질문 시 논문 관련 질문으로 유도
사이드바에서 지난 대화 기록 확인
새로운 대화 창 생성 가능
논문 요약 제공




주요 기능

02

메인화면

이메일을 통한 회원 가입 및 로그인
사용자 정보 수집에 관한 약관 동의 사항 제시
3단계로 구성된 사용자 가이드 제공



기술 스택




챗봇 / 사이드바 / 검색

질문 시 AI가 질문이나 요청에 응답
논문 관련되지 않은 질문 시 논문 관련
질문으로 유도
사이드바에서 지난 대화 기록 확인
새로운 대화 창 생성 가능
논문 요약 제공



마이페이지

작성 중인 논문 PDF 형식 업로드
작성 중인 논문 목록화
맞춤법 교정/검사
작성 중인 논문 작성 진행률 제공
논문을 AI가 분석하여 정보 제공



시장 분석

03

시장 분석

연간 논문 수 추이 (2018-2022)

AI로 세계 기술 특허 - 논문 찾는다는... '테크-GPT' 구축 본격화

2022년 10월 기준, AI 관련 논문 수 추이 (2018-2022)

2022년 10월 기준, AI 관련 논문 수 추이 (2018-2022)

시장 분석 : 유사 사례

R Discovery



학술 연구 논문 검색 지원
학술 기사, 과학 저널, 오픈 액세스 기사 및
최신 학술 연구에 대한 최신 정보 업데이트
음성 지원
번역 지원 (영어, 일본어)

DBpia



논문 검색/저장: 400만 편 이상의 학술 자료 제공
PDF 뷰어: 논문 열람, 검색, 다운로드 지원
논문과 서제 정보 정리 및 기기 간 동기화
DBpia 제휴 기관 소속 시 무료 논문 열람 가능
소속 인증 후 90일 유지, 간편 재인증
한국심리학회, 대한건축학회 등 3000여 학술기관 콘텐츠
열람 가능.

SWOT 분석

04

SWOT 분석

핵심 전략

사용자 친화적 기능으로 초기 사용자 확보
데이터 품질과 보안을 강화해 신뢰 구축
경쟁 서비스와 차별화된 추천 및 요약 기능 개발
글로벌 시장을 겨냥한 다국어 지원과 협력 확장

SWOT 분석

| | |
|---|---|
| S AI 기반 자연어 검색으로 기존 키워드 검색보다 정확 관리 사용자 맞춤 추천 시스템으로 개인화된 학술 자료 제공 직관적이고 사용자 친화적인 UI/UX | W 개발 및 유지 보수 비용의 과다 유료 학술 자료 접근 제한으로 데이터 베이스 확장에 어려움 초기 사용자 확보와 신뢰도 구축이 관 |
| O AI 학술 도구 수요의 증가와 학습/연구 환경의 디지털화 글로벌 시장 진출 가능성 대학/연구소와의 파트너십 및 협업 기회 | T 구글 학술검색 등 경쟁 서비스와의 차별화 필요 데이터 보안 이슈와 개인정보 보호 규제 강화 외부 API 의존성으로 서비스 제한 위험 |



향후 계획

05



향후 계획



수익 시스템

primieum 멤버십을 이용한 수익 창출

- > 고급 검색 기능
- > 멤버십 전용 커뮤니티
- > 개인화 서비스



향후 계획

SCI 등재 논문 데이터 연동하여 현재
보유하고 있는 국내 논문의 데이터 뿐만
아니라 더 많은 데이터 제공