

한성대학교
컴퓨터공학부
2024-2 SW프리캡스톤디자인
최종보고서

Book4U

팀원: 강경민, 김선준, 문기업, 이희찬

담당 교수: 김성동

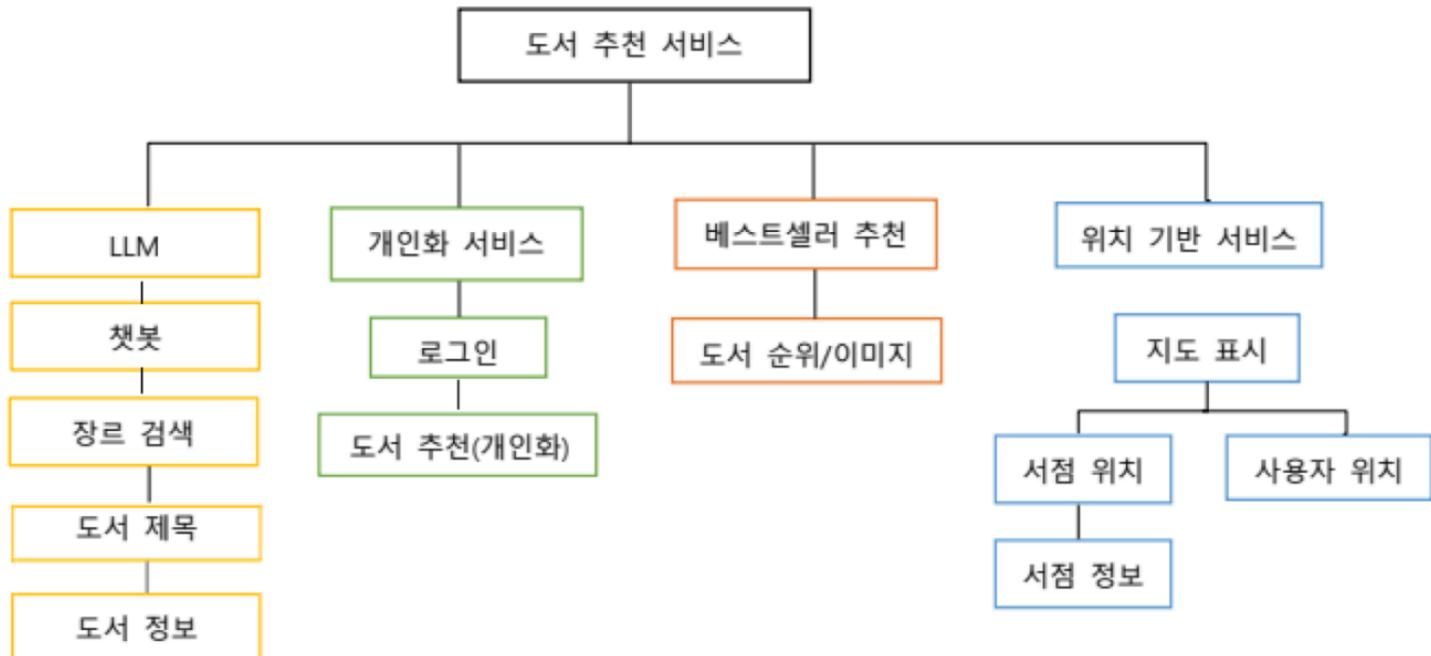
1. 프로젝트 계획서

1. 프로젝트 개요

1.1 프로젝트 이름
Book4U
1.2 문제점의 문
사용자의 연령, 독서 취향 등 다양한 개인적 특성을 반영하여 선호하는 장르와 주제를 기반으로 도서를 제안
1.3 프로젝트 배경 및 필요성
<p>-프로젝트 배경-</p> <p>최근 몇 년간 인공지능(AI)과 머신러닝(ML) 기술의 발전으로 인해, 다양한 분야에서 사용자 맞춤형 서비스가 활성화되고 있습니다.</p> <p>특히 도서 추천 분야에서는 사용자의 관심사, 나이, 취향에 맞는 도서를 추천 받는 것이 중요해졌습니다. 하지만, 기존의 도서 추천 서비스들은 사용자의 관심사, 나이, 취향에 맞춘 개인화된 추천에 한계가 있으며, 특히 특정 장르에 대한 세분화된 추천 서비스의 부재가 큰 문제로 지적되고 있기에, 사용자 맞춤형 도서 추천 서비스를 기획하게 되었습니다.</p> <p>-프로젝트 필요성-</p> <p>다양한 나이대와 취향을 반영한 맞춤형 도서 추천 시스템은 독서 문화 활성화에 중요한 역할을 할 수 있습니다.</p> <p>기존의 도서 추천 서비스는 제한된 데이터를 기반으로 하기 때문에,</p> <p>사용자 개개인의 나이, 취향, 관심 장르를 정확히 반영하는 데 한계가 있습니다.</p> <p>그러나 LLM 기술을 활용한 도서 추천 시스템은 이러한 한계를 극복할 수 있습니다.</p> <p>LLM은 방대한 양의 텍스트 데이터를 분석하고,</p> <p>사용자의 대화 패턴과 선호도를 깊이 이해하여 보다 개인화된 도서 추천을 제공할 수 있습니다. LLM은 사용자가 명시적으로 언급하지 않은 취향이나 관심사를 파악하는 능력을 갖추고 있으며,</p> <p>이를 통해 독자들은 자신에게 딱 맞는 책을 더 쉽게 찾을 수 있게 되며, 도서 선택의 폭이 크게 넓어집니다. LLM 기반의 추천 시스템은 사용자의 관심사를 지속적으로 반영하며,</p> <p>독자가 새로운 도서와 장르를 발견하고 흥미를 잃지 않도록 도와줍니다.</p> <p>이런 시스템은 전 세대에 걸쳐 독서 참여를 증진시키고,</p> <p>독서 문화의 활성화에 기여할 수 있는 중요한 도구가 될 것입니다.</p>

2. 기능 구조도(Functional Decomposition Diagram)

2.1 기능 구조도



2.2 기능 설명 – 각 기능 간단 설명

2.2.1 LLM

- 챗봇: 질문 가능한 Prompt 형식, 장르 추천 질문
- 장르 검색: 오케스트트로 LLM API 사용, 사용자가 특정 장르를 검색
- 나이대 선택: 나이대를 표시하는 체크박스 출력, 해당하는 나이대 선택
- 도서 제목: 알라딘 API 사용, 도서 제목 클릭 시 도서 정보 출력
-> (이미지, 상품명, 평점, 장르, 저자, 출판사, 출간일, 페이지 수, 정가, 줄거리)

2.2.2 개인화 서비스

- 로그인: 데이터베이스 사용, 사용자 개개인의 정보, 대화내용을 저장
- 대화기록: 로그인 정보를 통해 사용자 각각의 대화기록 저장, 출력 가능
- 도서 추천(개인화): 사용자 조회 도서 정보와 추천도서 목록을 기반으로 도서 추천

2.2.3 베스트셀러 추천

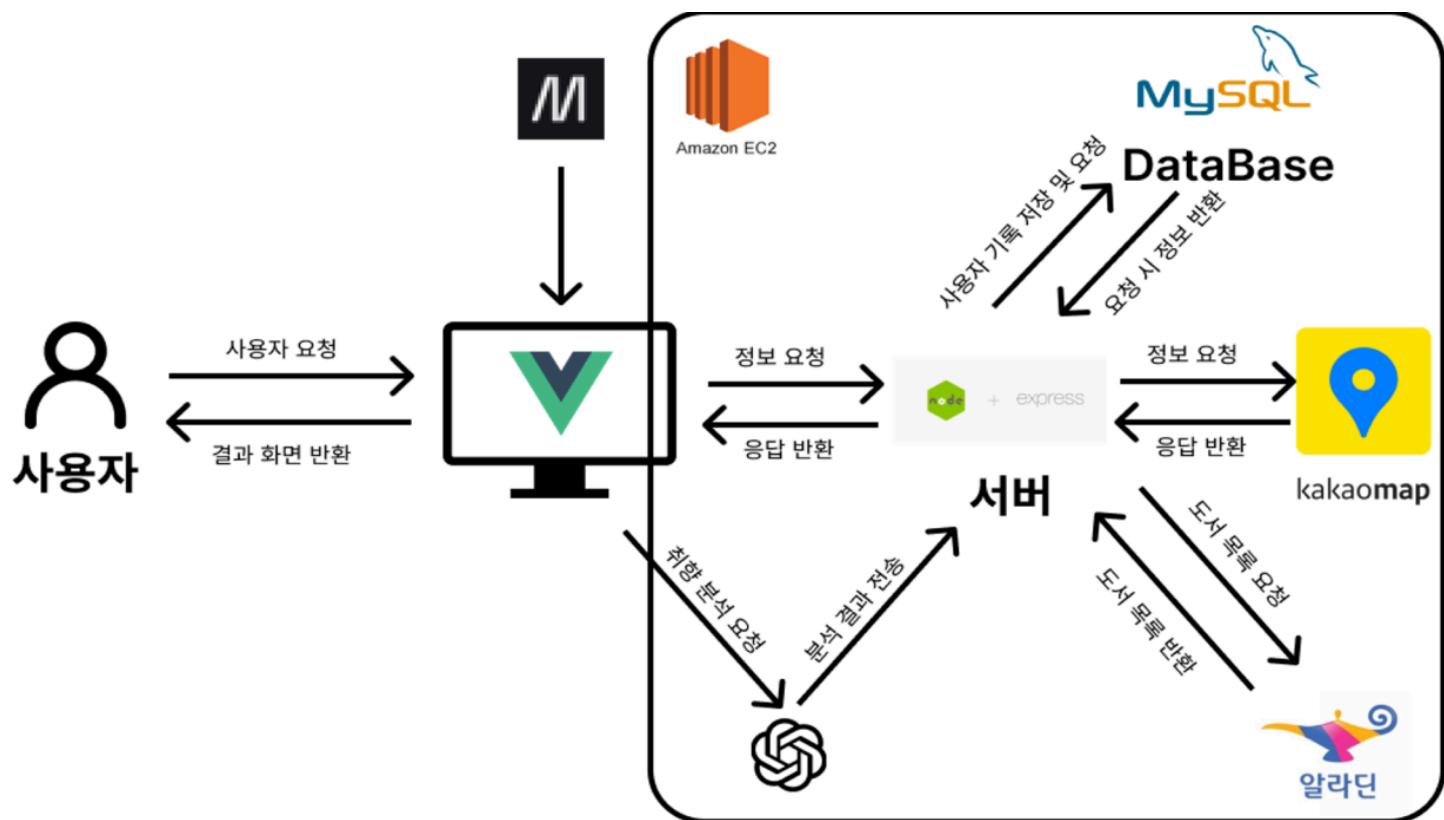
- 도서 순위/이미지: 알라딘 API 사용, 주간 베스트셀러 정보를 얻어와 순위와 이미지 출력

2.2.4 서점 위치 제공

- 지도 표시: 카카오맵 API를 통해 지도를 구현
- 사용자 위치: Geolocation API를 사용, 현재 위치를 중심으로 설정, 표시
- 서점 위치: 현재 위치 근처의 서점을 검색, 표시
- 서점 정보: 오버레이를 통해 서점의 정보를 출력

3. 시스템 구조도

3.1 시스템 구조도



3.2 시스템 구성 요소 설명

구성요소	설명	개발 환경 및 구현 방법
DB	회원 정보 관리	MySQL DBMS를 통해, 회원 정보 관리 데이터베이스 구축 예정
서버	API 통합 및 데이터 처리	Node.js를 기반으로 하고, Express.js 프레임워크를 사용하여 서버 구현. 카카오맵 API(지도 구현), 알라딘 API(도서 정보), 오피스트로 LLM API(챗봇 시스템)를 통합하고, DB와 연동하여 개개인의 데이터를 관리.
프론트 프레임워크 및 UI	사용자 인터페이스 및 상호작용	Vue.js를 기반으로 UI/UX 구현. Vuetify를 사용하여 일관된 디자인과 반응형 컴포넌트 구현. Geolocation API를 사용하여, 현재 위치 정보 획득. 서버로부터 받은 API 관련 정보를 화면에 출력
개발 환경 및 빌드 도구	프로젝트 구조, 개발 서버, 빌드 프로세스	Vite를 사용하여, 프로젝트 구조화 및 개발 서버 구동. 라우팅은 Vue Router를 통해 구현

4. 일정 계획

주차	마일스톤/작업	설명
5주 차	프로젝트 계획서 작성	프로젝트 목표 설정 및 팀 역할 배정
6주 차	웹 페이지 디자인(피그마)	UI/UX 디자인
7주 차	기본 구조 개발 (FE & BE)	FE/BE 기본 구조 구현, API 설계 및 초기 구현
8주 차	주요 기능 개발	FE/BE 주요 기능 구현, API 연동 및 단위 테스트
9주 차	FE/BE 연동 & 중간 발표 준비	모듈 간 연동, 통합 테스트 및 오류 수정
10주 차	중간 발표 및 피드백	개발한 기능을 토대로 발표 및 피드백
11주 차	UI/UX 개선 및 성능 최적화	피드백 기반 UI/UX 개선, 성능 최적화 작업
12주 차	추가 기능 구현 및 버그 수정	피드백 기반 추가 기능 구현, 알려진 버그 수정
13주 차	최종 테스트 및 버그 수정	전체 시스템 테스트, 개선 사항 점검
14주 차	최종 결과물 및 결과 발표	최종 결과물 시연 및 프로젝트 결과 발표

5. 역할 분담

5.1 김선준	-UI/UX 설계 및 구현- Vue.js를 사용한 사용자 인터페이스 개발, Vuetify를 활용한 반응형 디자인 구현, 프론트엔드 성능 최적화
5.2 강경민	-기능 구현 및 API 연동- 백엔드 API와의 통신 로직 구현, 상태 관리 및 데이터 흐름 설계, 프론트엔드 단위 테스트 및 통합 테스트 수행
5.3 문기업	-서버 개발 및 관리- Node.js 및 Express.js를 사용한 서버 구축, 외부 API (카카오맵, 알라딘, 오캐스트로 LLM 연동)
5.4 이희찬	-데이터베이스 설계 및 관리- MySQL 데이터베이스 설계 및 구현, 데이터 모델링 및 스키마 최적화, 데이터베이스 쿼리 최적화, 백엔드 단위 테스트 및 통합 테스트 지원

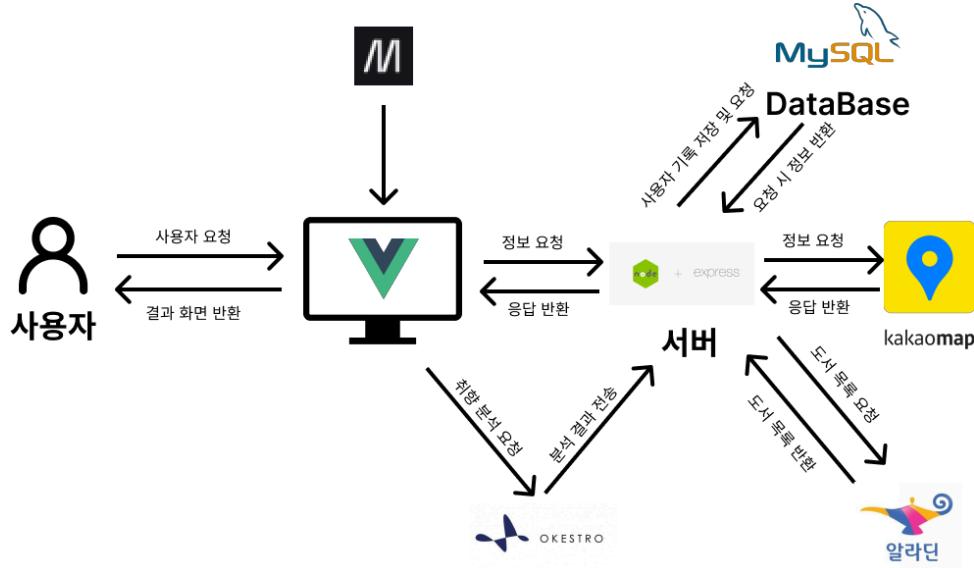
2. SW 상세 설계서

1. 문서 개요

- 목적:
이 설계서의 목적은 프로젝트의 전체 시스템 구조와 주요 구성 요소에 대한 상세한 정보를 제공하는 것입니다. 이 문서에서는 시스템의 아키텍처, 모듈 설계, 데이터베이스 설계, 사용자 인터페이스, API 설계 및 보안 방안 등을 포함하여 팀원 간 명확한 지침을 가지고 프로젝트를 진행할 수 있도록 지원합니다.
- 범위: 해당 설계서는 웹 애플리케이션의 전반적인 설계를 다룹니다.
 - 시스템 구조: 시스템의 전체 구조와 컴포넌트 간의 관계를 설명합니다.
 - 모듈 상세 설계: 각 모듈의 기능 및 역할을 상세히 기술합니다.
 - 데이터베이스 설계: 데이터베이스 구조와 테이블 관계를 설명합니다.
 - 사용자 인터페이스 설계: UI/UX 요소 및 디자인 원칙을 다루며, 사용자 경험을 극대화하기 위한 인터페이스 구성 및 상호작용 방식을 설명합니다.
 - API 설계: 시스템 간 데이터 통신을 위한 API의 정의 및 사용 방법을 명시합니다.
 - 보안 설계: 시스템의 보안 요구 사항과 이를 충족하기 위한 방안을 설명합니다.
- 참조 문서: 팀 4 프로젝트 계획서

2. 시스템 구조

2.1 전체 시스템 구조도



2.2 주요 컴포넌트 설명

데이터베이스 (MySQL) - 회원 정보 관리

- MySQL DBMS를 사용하여 회원 정보 관리 데이터베이스를 구축합니다. 사용자 데이터 및 서비스와 관련된 다양한 정보를 저장하고 관리합니다. 이 데이터는 서버와 연동하여 실시간으로 사용되며, 각 사용자의 개인화된 정보와 활동 기록을 저장합니다.

서버 (Node.js + Express) - API 통합 및 데이터 처리

- Node.js와 Express.js 프레임워크를 사용하여 서버를 구축합니다. 이 서버는 다양한 외부 API와 통합되며, 데이터 처리를 담당합니다. 주요 통합 API로는 카카오맵 API (지도 서비스), 알라딘 API (도서 정보 서비스), 그리고 오케스트로 LLM API (챗봇 서비스)가 있습니다. 서버는 이들 API와 상호작용하며, 데이터베이스와 연동하여 사용자 맞춤형 데이터를 제공합니다.

프론트엔드 프레임워크 및 UI (Vue) - 사용자 인터페이스 및 상호작용

- Vue.js를 기반으로 사용자가 직접 상호작용하는 UI/UX를 구현합니다. Vuetify를 사용하여 일관된 디자인과 반응형 컴포넌트를 구현함으로써, 다양한 디바이스에서 최적화된 사용자 경험을 제공합니다. 또한 Geolocation API를 활용해 사용자의 현재 위치 정보를 획득하고, 이를 서버로 전송하여 API 관련 정보를 화면에 출력합니다.

개발 환경 및 빌드 도구 (Vite) - 프로젝트 구조 및 개발 서버 관리

- Vite를 사용하여 프로젝트를 구조화하고, 개발 서버를 효율적으로 운영합니다. 빌드 과정에서 빠른 피드백을 제공하며, Vue Router를 통해 라우팅을 설정해 다양한 페이지 및 기능 간의 이동을 원활하게 처리합니다.

2.3 인터페이스 정의

프론트엔드 (Vue) ↔ 서버 (Node.js + Express)

- 통신 프로토콜: HTTP/HTTPS
- 데이터 형식: JSON
- 주요 요청:
 - GET /api/books: 알라딘 API에서 받아온 도서 정보를 서버에서 제공하여 클라이언트로 전달.
- 응답: API 요청에 대한 응답으로 JSON 형식의 데이터 반환.
- 인증:
 - 사용자의 비밀번호는 서버에서 `crypto` 모듈을 사용하여 해시화하여 저장합니다.
 - 사용자가 로그인할 때 입력한 비밀번호를 해시화하여 데이터베이스에 저장된 해시값과 비교하여 인증을 진행합니다.
 - 인증이 성공하면 `crypto.randomBytes()`를 사용하여 생성한 고유한 인증 토큰을 발급하고, 이를 클라이언트에게 반환합니다.
 - 클라이언트는 이후 요청 시 이 인증 토큰을 사용하여 서버에 인증된 요청을 보냅니다.

서버 (Node.js + Express) ↔ 데이터베이스 (MySQL)

- 통신 프로토콜: MySQL 프로토콜 (TCP/IP)
- 데이터 형식: SQL 쿼리 및 결과 집합
- 주요 작업:
 - 회원 데이터 관리: 사용자 등록, 로그인, 사용자 정보 업데이트 등을 위한 SQL 쿼리 사용.
 - 도서 저장: 사용자가 조회한 도서 목록을 MySQL에 저장
- 에러 처리: 데이터베이스 접근 중 예상 발생 시 서버는 오류 메시지를 클라이언트에 반환.

서버 ↔ 외부 API (카카오맵, 알라딘, 오케스트로)

- 카카오맵 API:
 - 통신 프로토콜: HTTP/HTTPS
 - 요청: 클라이언트에서 전달한 위치 데이터를 기반으로 지도 데이터를 요청.
 - 응답: 지도 위치 데이터 반환 (JSON 형식).
- 알라딘 API:
 - 통신 프로토콜: HTTP/HTTPS
 - 요청: 도서 정보 요청 (베스트셀러 목록, 특정 도서 상세 정보)
 - 응답: 도서 정보가 포함된 JSON 데이터를 반환.
- 오케스트로 LLM API:
 - 통신 프로토콜: HTTP/HTTPS
 - 요청: 사용자 입력에 대한 챗봇 응답을 위해 데이터 전달
 - 응답: 챗봇의 응답 메시지 데이터 반환 (JSON 형식)

프론트엔드 (Vue) ↔ 사용자 기기 (Geolocation API)

- 통신 방식: 브라우저 내장 API 호출
- 데이터 형식: 위치 데이터 (위도, 경도)

- 작업: 사용자가 요청 시 현재 위치 정보를 수집하여 서버에 전달. 위치 정보는 JSON 형식으로 포맷 후 서버에 전달됩니다.
- 보안: 위치 정보 사용 시 사용자 동의를 받도록 설정

3. 모듈 상세 설계

3.1 사용자 정보 관리 모듈

- 기능: 사용자 등록, 로그인,
- 입력: **id, password**, 닉네임, 생년월일, 이메일
- 출력: 회원가입 성공 여부, 로그인 토큰(?)
- 프로세스 흐름:
 1. 사용자가 회원 가입 정보 제출
 2. 회원 정보를 DB에 저장
 3. Node.js 내장 모듈 **crypto**를 활용하여 비밀번호 암호화
 4. 이메일, 비밀번호가 입력되면 사전에 생성된 Salt 값을 꺼내 비밀번호 암호화, DB내의 비밀번호와 비교
- 주요 컴포넌트 및 메서드:
 - **Login** 컴포넌트: 사용자가 로그인 아이콘을 눌렀을 때 이동하는 페이지. 해당 페이지에서 사용자의 로그인이 진행되고, 정상적으로 로그인 시 메인 화면으로 이동한다.
 - **SignUp** 컴포넌트 : 사용자의 회원 가입 시 표시되는 페이지. 해당 페이지를 통해 사용자의 **ID**, 비밀번호, 닉네임, 생년월일, 이메일을 데이터베이스에 저장한다.
 - **saveUserInfo({ userInfo })**: 사용자 정보를 데이터베이스에 저장
 - **encryptPassword({passwd})**: 사용자 비밀번호를 암호화
 - **verifyUser({ id, passwd })**: 사용자 로그인 시 데이터베이스에 저장된 정보와 일치하는지 확인
 - **decryptPassword(passwd)**: verifyUser 메소드에서 사용될 비밀번호 복호화 메소드

3.2 도서 정보 관리 모듈 (**Book Management module**)

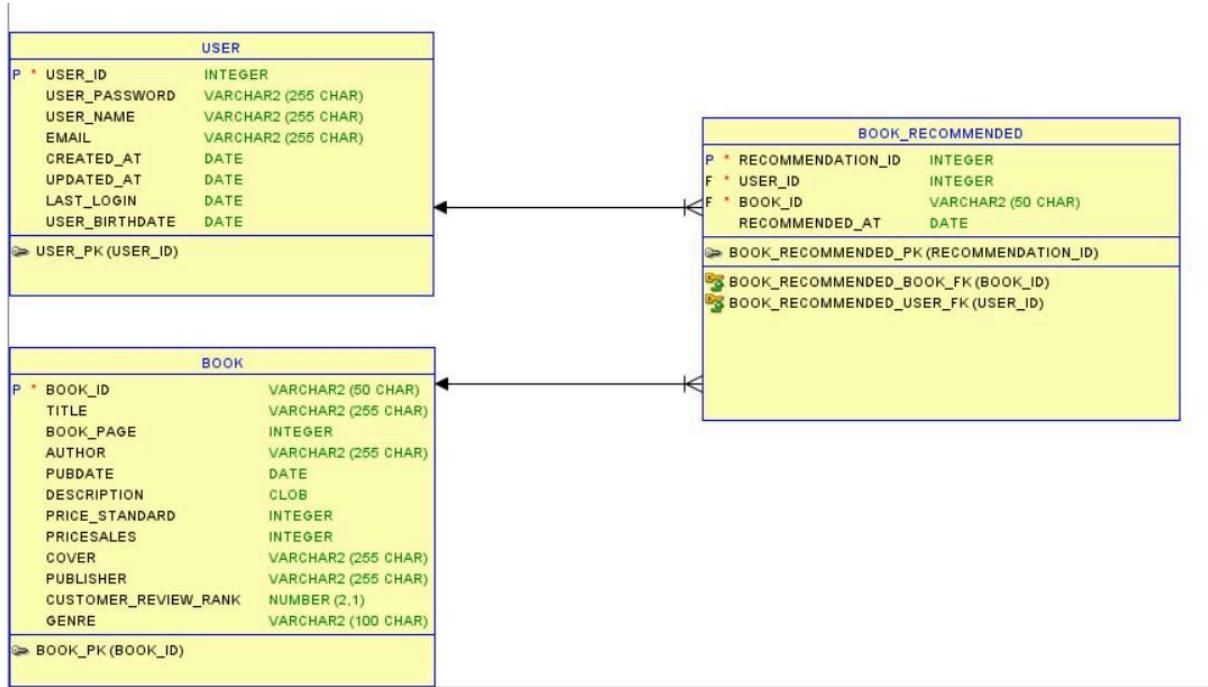
- 기능: LLM에서 제공된 도서 정보 저장, 도서 정보 조회
- 입력: 도서 정보(**BOOK_ID**, **TITLE**, 페이지 수, **AUTHOR**, 출판일, 설명, 정가, 판매가, 커버 이미지, 출판사, 평점, 장르, 조회일자)
- 출력(**Output**): 도서정보 (제목, 장르, 평점, 출판일, 저자 등등...), 상품 상세페이지 링크
- 프로세스 흐름:
 1. 사용자가 질문 입력하여 LLM에서 반환된 도서 정보를 DB에 저장
 2. 주천된 도서정보와 상세 페이지 링크 제공
 3. 사용자가 자신이 추천 받은 도서 목록을 조회
- 주요 컴포넌트 및 메서드:
 - **saveBookInfo({bookInfo})** : 도서정보 DB로 저장
 - **getBookInfo()** : book 객체 반환
 - **getBookList()** : 저장된 book리스트 반환

3.3 지도 출력 모듈

- 기능: kakao map api와 Geolocation api를 통해 지도와 사용자의 위치, 서점 위치 등을 출력
- 출력(Output): 지도화면 출력 (현재 사용자 위치, 주변 서점 위치 정보)
- 프로세스 흐름:
 1. 컴포넌트 마운트 (카카오 지도 API를 로드)
 2. 현재 위치 가져오기
 3. 지도 생성 (지도를 생성하고 현재 위치에 마커 표시)
 4. 장소 검색 (사용자 근처 서점 위치를 검색 후 출력)
 5. 검색 결과 처리 (API 결과를 처리하여 장소를 찾은 경우 마커 표시 준비)
 6. 마커 표시 (각 장소에 마커 생성)
 7. 오버레이 생성 (마커 클릭 시 해당 장소 오버레이 표시 -> 서점 관련 정보 표시)
 8. 현재 위치 마커 표시
 9. 오버레이 닫기 (X버튼을 클릭 시 오버레이 종료)
- 주요 컴포넌트 및 메서드:
 1. 지도 초기화 및 생성 (**mounted**와 **initMap**)
mounted 라이프사이클 흐름에서는 페이지가 로드될 때 카카오맵 SDK를 비동기로 로드하고, 로드가 완료되면 **initMap** 메서드를 호출하여 지도를 초기화
 2. 지도 생성 (**createMap**)
카카오맵 객체를 생성, **kakao.maps.LatLng**를 사용해 지도의 중심을 설정. 이후 **displayCurrentLocationImage**로 현재 위치를 지도에 표시
 3. 장소 검색 (**searchPlaces**와 **placesSearchCB**)
searchPlaces: **kakao.maps.services.Places** 객체를 사용하여 **keywords**에 포함된 모든 키워드(서점, 문고)에 대해 지도 내에서 장소를 검색
placesSearchCB: 검색이 성공하면 **displayMarker** 메서드를 호출하여 검색 결과로 나온 장소를 지도에 표시
 4. 현재 위치 표시 (**displayCurrentLocationImage**)
displayCurrentLocationImage: 사용자의 현재 위치를 지도의 특정 이미지(**homeImage**)로 표시하는 메서드.

4. 데이터베이스 설계

4.1 ER 다이어그램



[ERD 요약]

- USER 테이블은 사용자의 정보를 저장한다.
- BOOK 테이블은 책의 정보를 저장한다.
- BOOK_RECOMMENDED 테이블은 사용자에게 추천한 책 정보를 저장한다.

4.2 테이블 정의

- 테이블명 : USER
 - 컬럼 :
 - USER_ID(PK, INT, AUTO_INCREMENT)
 - USER_PASSWORD(VARCHAR)
 - USER_NAME(VARCHAR)
 - EMAIL(VARCHAR, UNIQUE)
 - CREATED_AT(DATETIME, CURRENT_TIMESTAMP)
 - UPDATED_AT(DATETIME, CURRENT_TIMESTAMP)
 - LAST_LOGIN(DATETIME)
 - USER_BIRTHDATE(DATE)
- 테이블명 : BOOK
 - 컬럼:
 - BOOK_ID(PK, VARCHAR)
 - TITLE(VARCHAR)
 - BOOK_PAGE(INT)
 - AUTHOR(VARCHAR)
 - AUTHOR(DATE)
 - DESCRIPTION(TEXT)
 - PRICE_STANDARD(INT)
 - PRICESALES(VARCHAR)
 - COVER(INT)
 - PUBLISHER(VARCHAR)
 - CUSTOMER REVIEW RANK(DECIMAL(2,1))
 - GENRE(VARCHAR)
- 테이블명: RECOMMENDED_BOOK
 - 컬럼:
 - RECOMMENDATION_ID(PK, INT, AUTO_INCREMENT)
 - USER_ID(USER FK, INT)
 - BOOK_ID(BOOK FK, INT)
 - RECOMMENDED_AT(DATETIME, CURRENT_TIMESTAMP)

4.3 테이블 간 관계

1. USER와 BOOK_RECOMMENDED:

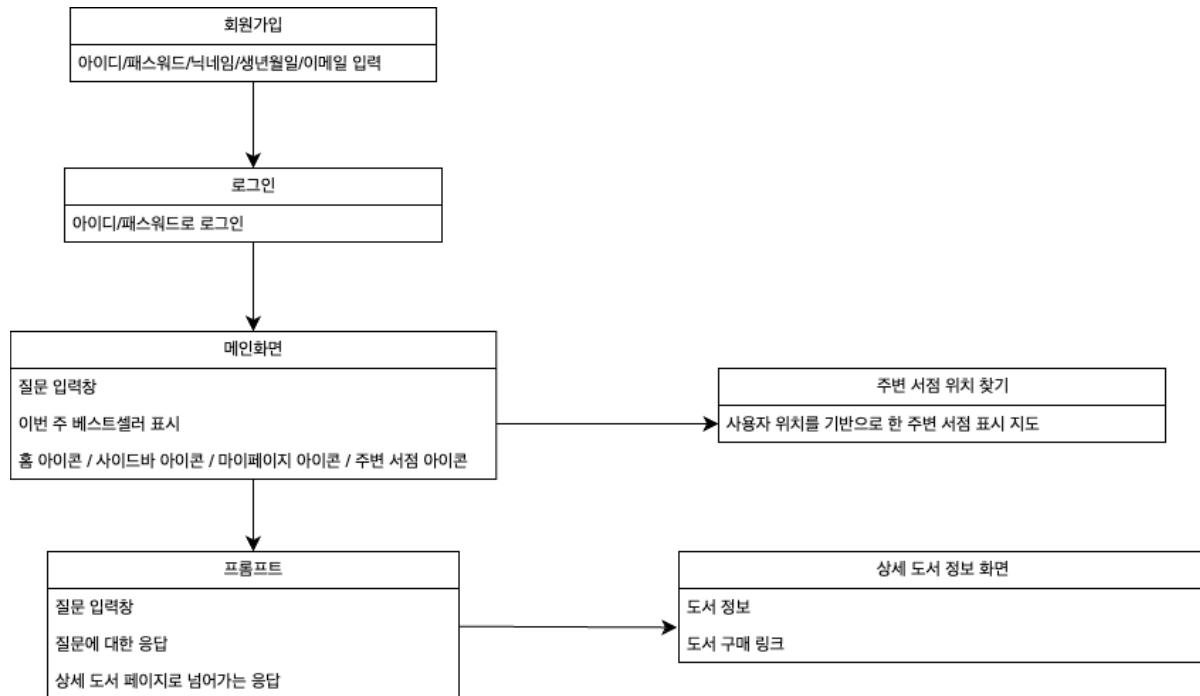
- 관계: 일대다 (One-to-Many)
- 설명: 사용자는 여러 가지의 도서들을 추천을 받을 수 있다. 따라서, USER 테이블의 USER_ID가 BOOK_RECOMMENDED 테이블에서 여러 번 사용된다.

2. BOOK과 BOOK_RECOMMENDED:

- 관계: 일대다 (One-to-Many)
- 설명: 도서는 여러 사용자에게 추천을 해줄 수 있다. 따라서 BOOK 테이블의 BOOK_ID가 BOOK_RECOMMENDED 테이블에서 여러번 사용된다.

5. 사용자 인터페이스 설계

5.1 화면 구조도



5.2 주요 화면 설계

<메인화면>



도서와 관련된 정보를 검색해주세요



<사이트바>



나미야 잡화점의 기적

요청 일시: 2024-10-14

미움 받을 용기

요청 일시: 2024-10-13

인간 실격

요청 일시: 2024-10-11

변신

요청 일시: 2024-09-28

나의 라임오렌지 나무

요청 일시: 2024-09-24



판타지 소설을 검색해줘



판타지 소설을 찾으시는군요!
나이대를 선택해주세요.

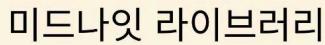
- | | |
|-----|---|
| 10대 | ✓ |
| 20대 | ✓ |
| 30대 | ✓ |
| 40대 | ✓ |
| 그 외 | ✓ |

1. 나미야 잡화점의 기적
2. 미드나잇 라이브러리
3. 바다가 들리는 편의점2

메시지를 입력해주세요



<상세 응답 화면>





제목: 미드나잇 라이브러리

★10

장르: 판타지

작가: 매트 헤이그

출판사: 인플루엔셜

발매일: 2021. 04. 28

페이지 수: 406

정가: 14300원

-줄거리-

주인공 노라 시드는 자신의 인생에 불만을 가지고 집에서 자살을 시도했고
눈을 뜨니 사방에 안개가 깔린 장소에 와 있었다.

그곳에는 작은 슈퍼마켓 크기의 도서관이 있는데 시간이 00:00:00에서 흐르지 않는다.

초록색의 책들이 가득한 도서관에는 생전 사서였던 엘름 부인이 있다.

이곳에서 엘름 부인은 노라의 생각을 읽을 수 있다.

엘름 부인이 말하는 바에 의하면 삶과 죽음 사이에는 거쳐가는 중간 지대가 있는데,
이것은 사람마다 모두 다르다고 한다. 노라의 경우 '도서관'인 것이다.

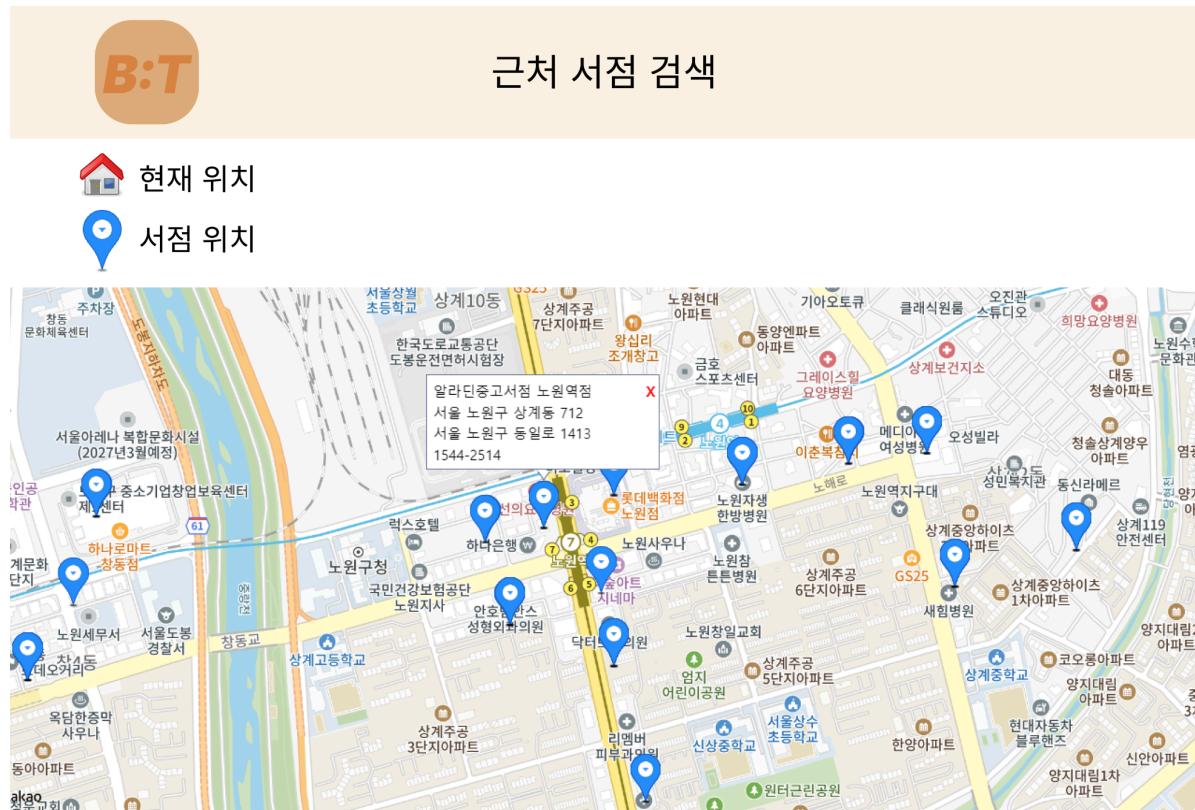
이곳을 '자정의 도서관(Midnight Library)'이라고 부르고,

자정의 도서관에 꽂힌 책들에는 내가 살 수도 있었던 삶을 살아볼 기회가 담겨 있다.

따라서 책을 펼치면 그 선택을 한 삶을 경험할 수 있다.

여러 가지 삶들을 경험해 본 노라는 결국 자신의 원래 삶이 가장 행복하고 가치 있다는 것을 발견하고,
도서관이 붕괴되면서 노라는 현실로 돌아온다.

<주변 서점 지도 화면>



<로그인 및 회원가입>

회원가입

아이디

비밀번호

비밀번호 확인

닉네임



ID 또는 E-Mail



비밀번호

생년월일

 년 월 일

이메일

 @

Login

회원가입

가입하기

5.3 사용자 시나리오

- 회원가입 시나리오
 - 회원 페이지에 접속 시 로그인 페이지로 이동 후, 회원가입 버튼을 눌러 회원가입 진행. 회원가입 완료 후 로그인 페이지로 다시 이동. 로그인이 완료되면 메인 페이지로 전환
- 검색 시나리오
 - 웹페이지에 처음 접속 또는 재접속했을 경우, 메인 페이지의 입력창에 질문을 입력하여 요청
 - 질문을 계속 이어나가는 경우, 프롬프트 페이지의 입력창에 질문을 입력하여 요청
 - 장르를 기준으로 질문했을 때, 나이대를 선택할 수 있는 체크박스를 통해 추가 질문을 요구하고, 그에 맞는 응답을 전달.
 - 장르가 아닌 질문(ex: 특정 작가의 작품, 책 가격대, 책 페이지 수 등등)에 관련된 질문인 경우, 나이대 선택은 넘어가고 해당 질문에 대한 응답을 전달.
- 도서 상세 정보 열람 시나리오
 - 프롬프트 페이지에서 LLM의 응답으로 도서명에 링크가 생기고, 사용자가 도서에 대한 상세 정보를 열람을 원하는 경우 이 링크를 통해 상세 응답 화면 페이지로 이동하게 됨.
 - 상세 응답 화면 페이지에서는 해당 도서의 상품명, 페이지수, 장르, 저자, 출간일, 상품설명, 정가, 판매가, 커버 이미지, 출판사, 평점 등 출력
- 주변 서점 지도 확인 시나리오
 - 메인 페이지, 프롬프트 페이지, 도서 상세 페이지의 오른쪽 상단에 있는 마커 표시를 누르게 되면 사용자의 현재 위치를 기반으로 주변 서점에 대한 정보가 담긴 지도 페이지로 이동함.
 - 지도에서 표시된 서점 마크를 누르면 해당 서점의 위치 정보, 서점 이름 등의 정보를 확인 가능하게 됨.

6. API 설계 (API Design)

6.1 Geolocation API (현재 사용자 위치 정보 api)

- URL: /api/geolocation/current
- Method: GET
- Request Body: 현재 위치 조회(GET 요청) -> 본문 필요X

```
navigator.geolocation.getCurrentPosition((position) => {  
    doSomething(position.coords.latitude, position.coords.longitude);  
});
```

- Request:

- Response:

```
(position) => {  
    const lat = position.coords.latitude;  
    const lng = position.coords.longitude;  
    this.currentLocation = { lat, lng }; // 현재 위치 저장  
    this.createMap(lat, lng);  
}
```

- 예외 처리:

PERMISSION_DENIED : 페이지에 필요한 권한이 없어(예: 권한 정책에 의해 차단됨)
지리적 위치 정보 수집에 실패

POSITION_UNAVAILABLE : 하나 이상의 내부 위치 소스가 내부 오류를 반환했기
때문에 지리적 위치를 획득 실패

TIMEOUT : 정보를 얻기 전에 지리적 위치를 획득할 수 있는 시간초과

6.2 Kakaomap API

- URL: <https://map.kakao.com/link/map>/위도,경도
- URL Pattern: /link/map/위도,경도
- Method: GET
- Request Body: 본문 필요 없음 (GET 요청은 URL에 정보를 포함시킴)

```
<script type="text/javascript"
src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은 APP KEY"></script>
```

- Request:

- Response:

```
var container = document.getElementById('map'),
options = {
    center: new kakao.maps.LatLng(33.450701, 126.570667),
    level: 3
};

var map = new kakao.maps.Map(container, options);
```

- 에러 처리:

```
const script = document.createElement('script'); script.onload = () =>
kakao.maps.load(this.initMap); script.onerror = () => alert('카카오맵 API
로드에 실패했습니다. 인터넷 연결을 확인하고 다시 시도하세요.');
```

6.3 Aladin API

- URL: /api/ItemList.aspx
- Method: GET
- Request Body: 본문 필요 없음(GET 요청은 URL에 정보를 포함시킴)

```
axios.get('/api/ItemList.aspx', {
  params: {
    ttbkey: 'YOUR_TTB_KEY_HERE', // 실제 TTBKey를 입력하세요
    QueryType: 'Bestseller',
    MaxResults: 10,
    start: 1,
    SearchTarget: 'Book',
    output: 'js',
    Version: '20131101',
  },
});
```

● Request

● Response:

```
{
  "item": [
    {
      "isbn13": "9781234567890",
      "bestRank": "1",
      "title": "책 제목 예시",
      "cover": "http://example.com/cover.jpg"
    },
    {
      "isbn13": "9780987654321",
      "bestRank": "2",
      "title": "또 다른 책 제목",
      "cover": "http://example.com/cover2.jpg"
    }
    // 추가 책 항목...
  ]
}
```

● 에러 처리:

```
catch (error) {
  if (error.response) {
    console.error('응답 오류:', error.response.data);
```

```
    this.error = error.response.data.message || '알 수 없는 오류가  
발생했습니다.';  
} else if (error.request) {  
    console.error('응답을 받지 못했습니다:', error.request);  
    this.error = '서버에서 응답이 없습니다.';  
} else {  
    console.error('오류:', error.message);  
    this.error = '요청 중 오류가 발생했습니다.';  
}  
}
```

7. 보안 설계

7.1 인증 및 권한 관리

- 사용자 인증 방식
 - ID/비밀번호 기반 인증: 최소 8자리 이상의 강력한 비밀번호 정책(대소문자, 숫자, 특수문자 조합).
- 권한 관리
 - 최소 권한 원칙: 일반 사용자 기능외에 필요한 기능이 없기에 최소한의 사용자권한만 부여

7.2 데이터 암호화

• 비밀번호 보안

- **crypto** 모듈의 암호화 기능

1. salt 값 생성: crypto 모듈의 `createSalt` 함수를 통해 생성

```
const createSalt = async () => {
  const buf = await randomBytesPromise(64);

  return buf.toString("base64");
};
```

2. 비밀번호 암호화: crypto 모듈의 `pbkdf2` 함수를 통해 암호화 (인자: salt값, 해싱할 값, 해시 함수 반복 횟수, 해시 값 길이, 해시 알고리즘)

```
export const createHashedPassword = async (password) => {
  const salt = await createSalt();
  const key = await pbkdf2Promise(password, salt, 104906, 64, "sha512");
  const hashedPassword = key.toString("base64");

  return { hashedPassword, salt };
};
```

3. 비밀번호 검증: `verifyPassword` 함수를 통해 입력된 비번, 사용자 비번, salt값의 3가지 인자를 비교하여 검증

```
export const verifyPassword = async (password, userSalt, userPassword)
=> {
  const key = await pbkdf2Promise(password, userSalt, 99999, 64,
  "sha512");
  const hashedPassword = key.toString("base64");

  if (hashedPassword === userPassword) return true;
  return false;
};
```

7.3 보안 로그

- 암호화 성공
 - 암호화가 성공적으로 이루어진 경우, 어떤 작업이 수행되었는지, 어떤 사용자가 요청했는지 등의 정보를 기록
- 암호화 실패:
 - 암호화 작업이 실패했을 경우, 원인과 관련된 정보를 기록

8. 테스트 계획 (Test Plan)

- 유닛 테스트: 주요 서비스 클래스에 대한 유닛 테스트 계획 (예: 사용자 생성 기능의 유효성 검사)
- 통합 테스트: API 통합 테스트 계획
- 테스트 도구: Jest

8.1 단위 테스트 계획

- 목적: 개별 기능 및 모듈이 예상대로 동작하는지 확인합니다.
- 대상:
 - 사용자 정보 관리 모듈 (회원 가입, 로그인, 비밀번호 암호화 등)
 - 도서 정보 관리 모듈 (도서 정보 저장, 조회 등)
 - 지도 출력 모듈 (위치 기반 지도 서비스)
- 방법:
 - Jest 테스트 프레임워크를 사용하여 사용자 정보에 대한 저장, 조회, 암호화 등 사용자 정보 관리 모듈, 도서 정보에 대한 저장, 조회 등 도서 정보 관리 모듈, 사용자 위치 기반 지도 표시 서비스에 대한 단위 테스트를 진행

8.2 통합 테스트 계획

- 목적: 여러 모듈이 서로 원활하게 통합되어 동작하는지 확인합니다.
- 대상:
 - 프론트엔드와 백엔드 간의 API 통신 (**Vue ↔ Node.js**)
 - 백엔드와 데이터베이스 간의 통신 (**Node.js ↔ MySQL**)
 - 백엔드와 외부 API 간의 통신 (**Node.js ↔ 카카오맵, 알라딘, 오키스트로 API**)
 - 사용자가 도서를 검색하면 **LLM**을 통해 반환된 도서 정보가 **DB**에 저장되고, 그 정보를 클라이언트가 정상적으로 받아오는지 확인.
- 방법:
 - Supertest 라이브러리를 통해 통합 테스트 진행

8.3 엔드 투 엔드(E2E) 테스트

- 목적: 실제 사용자 관점에서 전체 시스템이 기대대로 동작하는지 확인합니다.
- 대상:
 - 사용자 회원가입, 로그인, 도서 검색, 도서 정보 조회, 지도 출력 등 사용자의 주요 시나리오.
- 방법:
 - supertest 라이브러리를 사용하여 브라우저 상에서 사용자 액션을 자동으로 실행하고, 결과가 기대와 일치하는지 확인합니다.
 - 예시: 사용자가 회원가입 후 로그인하고 도서 추천을 받는 시나리오가 원활히 작동하는지 테스트.

8.4 보안 테스트 (Security Test)

- 목적: 시스템이 잠재적인 보안 위협에 안전한지 확인합니다.
- 대상:
 - 사용자 인증 및 권한 관리 (**ID**/비밀번호 암호화, **JWT** 토큰 발급 및 관리)
 - 위치 정보 사용 시 사용자 동의 및 권한 관리

3. 최종 발표 자료

발표 대본: Book4U - 맞춤형 도서 추천 서비스

1. 시작 인사 및 문제 제시

안녕하세요, 오늘 저희 팀은 **Book4U**, 사용자 맞춤형 도서 추천 서비스에 대해 소개드리겠습니다.

독자들은 방대한 도서 선택지 속에서 자신에게 맞는 책을 찾는 데 어려움을 겪고 있습니다. 특히 바쁜 현대인들에게 책 선택의 시간과 에너지는 큰 부담으로 다가옵니다.

2. 해결책 제시

Book4U는 사용자 데이터를 기반으로 독자의 취향을 분석하여 맞춤형 도서 추천을 제공합니다.

- 사용자의 읽기 이력, 선호 장르, 작가, 평점 등을 종합적으로 분석해 최적화된 추천 결과를 도출합니다.
- 이로 인해 독자는 자신에게 꼭 맞는 책을 쉽고 빠르게 발견할 수 있습니다.

3. 추가 비즈니스 전략

- 광고 연계 추천: 출판사에서 광고하는 책을 사용자의 관심사와 연결해, 광고 도서가 효과적으로 노출되도록 합니다.
- 최상단 광고 노출: 배달의 민족의 슈퍼애드처럼, 광고한 서점이 추천 리스트의 최상단에 표시되도록 설계합니다.
- 구매 유도 링크: 도서 상세 페이지에서 제휴 사이트로 연결되는 구매 링크를 제공해, 사용자가 손쉽게 책을 구매하도록 합니다.

4. 시장 가능성

독서와 자기계발에 대한 관심은 꾸준히 증가하고 있습니다.

- 개인화 추천 서비스는 이미 영화나 음악 스트리밍에서 성공적인 사례를 보였으며, 독서 시장에서도 높은 가능성을 보입니다.
- 특히 전자책 및 오디오북 시장의 확장성과 연계하면 더 큰 성과를 기대할 수 있습니다.

5. 수익 모델

- 구독 서비스: 개인 맞춤형 추천을 정기적으로 제공하는 프리미엄 모델.
- 광고 수익: 출판사 및 서점 광고와 연계하여 추천 순위 상단에 노출.

제휴 서점 데이터베이스를 통해-> 기능구현(재고), 제휴서점 전용 아이콘을 통해 접근성 향상

- 판매 연계 수수료: 구매 링크를 통해 제휴 사이트에서 도서를 구매할 때 수익을 창출.

6. 확장 가능성

Book4U는 단순한 도서 추천을 넘어 전자책, 오디오북, 심지어 영화 및 드라마 추천으로도 확장 가능합니다.

- 추천 알고리즘의 높은 유연성을 활용해 다양한 콘텐츠 플랫폼과 연계할 수 있습니다.

7. 마무리 및 투자 요청

Book4U는 독자들이 도서 선택의 어려움을 해소하고, 출판사와 서점의 판매 증대를 도우며, 독서 문화 확산에 기여할 것입니다.

귀하의 투자로 Book4U가 전 세계 독자들에게 혁신적인 독서 경험을 제공할 수 있도록 힘을 더해주시십시오.

감사합니다.



BOOK4U :

플랫폼 수익 모델 및 전략

2071207 강경민

2071420 김선준

2071187 문기업

2071431 이희찬

목차



프로젝트 주제

맞춤형 도서 추천 시스템 개발

최근 몇 년간 인공지능을 활용한 다양한 사용자 맞춤 서비스가 개발되고 사용되고 있습니다.

그러나 전자기기와 IT 기술의 발전에 따라 현대인들의 독서량이 낮아지고 있는 추세입니다.

저희 팀은 이 주제에 초점을 맞추어, 책을 처음 접하는 사람들, 어떤 책을 읽어야 할지 고민하는 사람들, 새로운 장르의 책을 탐색하고 싶은 사용자를 위한 인공지능 기반 도서 추천 웹 서비스를 개발했습니다.



주제 설정 이유

독서 접근성 향상

개인 맞춤형 추천을 통해
다양한 책을 발견하고 독서
즐거움을 높입니다.

독서 흥미 유발

새로운 장르와 작가를
추천하여 독서 범위를
확장하고 흥미를 유발합니다.

개인화된 독서 경험

개인의 취향과 선호도를 반영하여
만족도 높은 독서 경험을 제공합니다.

주요 기능

1 회원가입

사용자는 이메일, 비밀번호, 관심 분야 등을 입력하여 회원가입을 진행합니다.

2 로그인

회원가입한 정보를 사용하여 시스템에 로그인합니다.

3 메인 페이지

베스트셀러 목록, 개인 맞춤형 도서 추천 등을 확인할 수 있습니다.

4 프롬프트(챗봇)

챗봇을 통해 도서 추천, 작가 정보, 책 관련 질문 등을 문의할 수 있습니다.

5 도서 상세 페이지

도서 상세 정보를 확인하고 좋아요를 눌러 좋아요 목록에 추가할 수 있습니다.

6 좋아요 목록

사용자가 좋아요를 누른 도서 목록을 확인할 수 있습니다.

7 지도

카카오맵을 이용하여 주변 서점을 찾고 위치 정보를 확인할 수 있습니다.

메인 페이지: 베스트 셀러, 추천도서

B:U

오늘은 어떤 책을 읽어볼까요?

이번 주 베스트셀러
Top 30

이희찬님을 위한 추천 도서

Developers

Kang Kyungmin Kim Sunjun Moon Kiup Lee Heechan

프롬프트 페이지

Book4U

이희찬 님 환영합니다!   

한강 작가의 책을 읽고 싶어

B:U 한강 작가의 작품은 감성적이고 깊이 있는 문체로 많은 독자들에게 사랑받고 있습니다. 다음은 한강 작가의 대표작들입니다:

1. [채식주의자](#) (The Vegetarian) - 한강
2. [소년이 온다](#) (Human Acts) - 한강
3. [흰](#) (The White Book) - 한강

이 책들은 한강 작가의 독특한 문체와 깊이 있는 주제를 잘 보여주는 작품들로, 한국 문학에 관심 있는 독자들에게 추천드립니다.

메시지를 입력해주세요. 

상세 페이지: 도서 상세 정보







채
식
주
의
자
한강
장편소설

채식주의자 (리마스터판) - 2024 노벨 문학상 수상작가

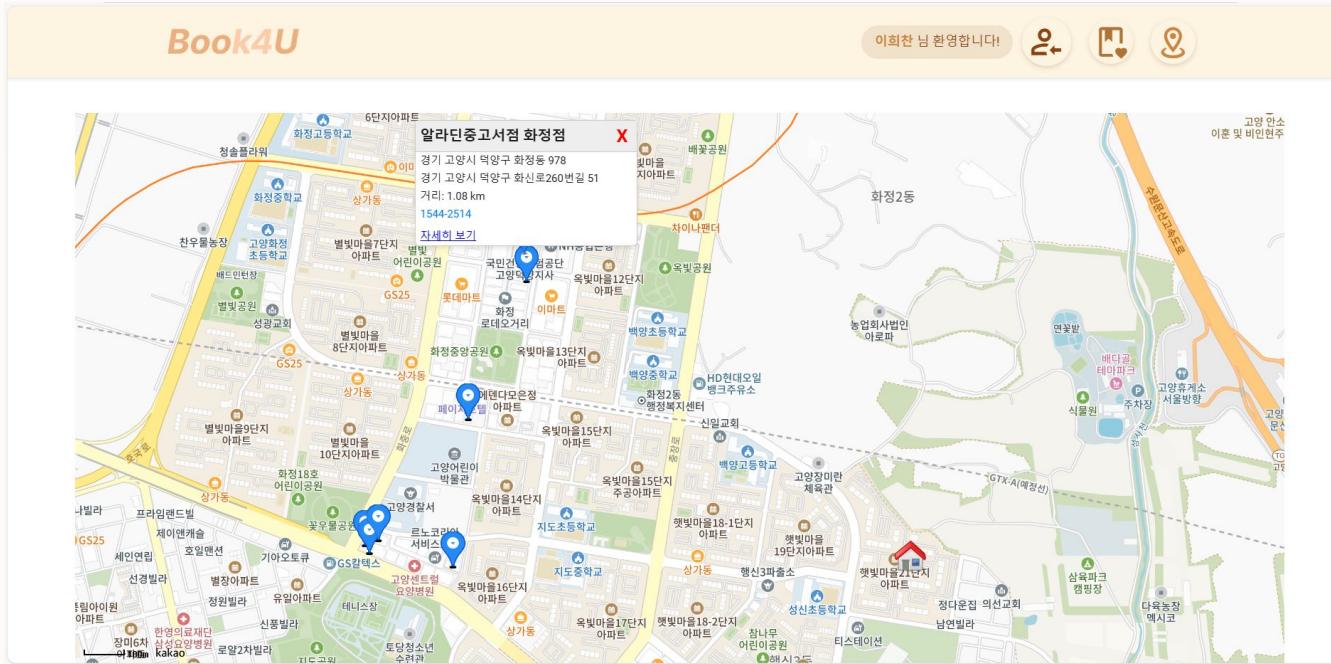
 원제/목:

 8
장르: 국내도서>소설/시/희곡>한국소설>2000년대 이후 한국소설
저자: 한강 (지은이)
ISBN: 9788936434595
페이지 수: 276
출간일: 2022-03-28
출판사: 창비
정가: 15000원
판매가: 13500원
[구매링크](#)

상품 설명

2016년 인터내셔널 부커상을 수상하며 한국문학의 입지를 한단계 확장시킨 한강의 장편소설. 상처받은 영혼의 고통과 식물적 상상력의 강렬한 결합을 정교한 구성과 흡인력 있는 문체로 보여주며 섬뜩한 아름다움의 미학을 한강만의 방식으로 완성한 역작이다.

인근 서점



A screenshot of the KakaoMap listing for Aladdin Middle School Hwajeong Branch. The listing includes a photo of stacked books, the store name (알라딘중고서점 화정점), a rating of 후기 3.8 (6), and 29 reviews. It also shows the address (경기 고양시 덕양구 화신로 260번길 51), operating hours (09:30 ~ 22:00), and a contact phone number (1544-2514 대표번호). The listing is updated on 2024.09.25.

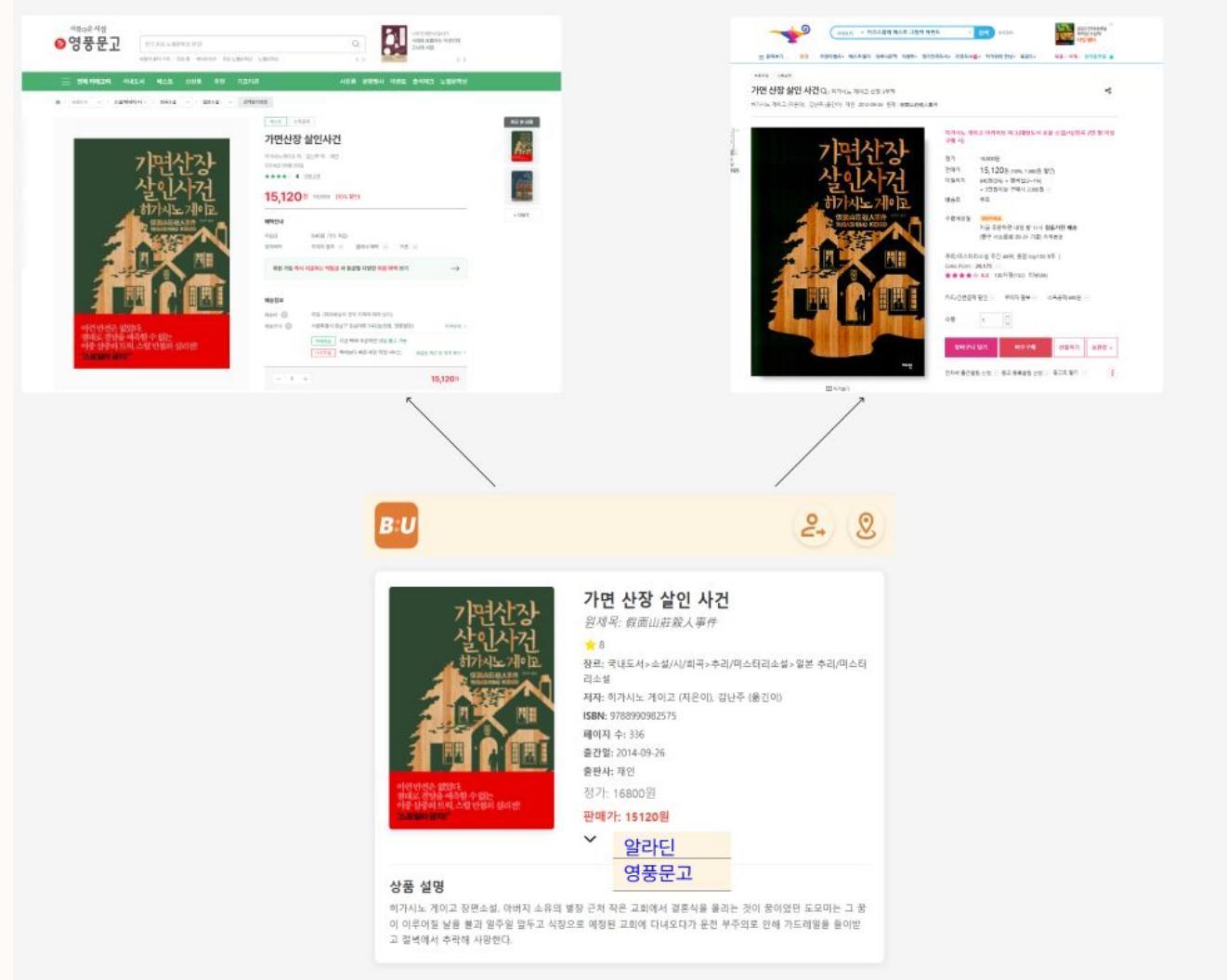
도서 상세 정보 페이지를 통한 수익 창출

제휴 확대

다양한 온라인 서점과의 제휴를 통해 구매링크를 추가하여 사용자에게 더 많은 구매 옵션, 구매에 대한 편의성을 제공할 계획입니다.

수익 창출

클릭 기반 수익(CPC) 및 구매 전환 기반 수익(CPA)을 창출할 수 있습니다. 또한, 제휴 서점의 경쟁력을 높여 플랫폼의 핵심 수익 모델로 자리 잡을 것입니다.



주변 서점 페이지 광고 수익화



주변 서점 광고

배달의민족과 유사한 광고 모델을
도입하여 주변 서점의 상단 노출 및
지도 표시를 통해 수익을 창출합니다.



유료 광고 기능

서점은 유료 광고 기능을 통해 더욱
돋보이는 홍보 효과를 얻을 수
있습니다.

이회찬 님 주변의 서점 목록

서점명	거리
교보문고	232m
영풍문고	347m
알라딘	434m
노원문고	518m
당산 책방	121m
천재문고	213m

B:U

이회찬 님 환영합니다!

250m kakao

사용자 맞춤 추천 기능과 출판사 협업

1

개선된 추천 시스템

사용자의 독서 기록과
선호도를 분석하여 더욱
정확하고 개인화된 도서 추천
서비스를 제공합니다.

2

출판사 제휴

출판사와의 협업을 통해 특정
도서를 효과적으로 홍보하고
광고 수익을 창출합니다.

이희찬님을 위한 추천 도서

Developers

Kang Kyungmin Kim Sunjun Moon Kiup Lee Heechan

프로젝트의 종합적 수익 모델



결론 및 기대 효과

지속 가능성

다각적인 수익 모델을 통해 플랫폼의 지속 가능한 성장을 확보합니다.

투자 수익률 보장

제휴사 및 광고주에게 투자 대비 높은 수익률을 보장합니다.

질문과 답변

