**종합설계PBL 최종 보고서**

프로젝트명: 트북이

(AI기반 노트북 추천 및 맞춤형 노트북 쇼핑몰)

팀명: 브레인스토밍

|  |  |  |
| --- | --- | --- |
| 전공 | 학번 | 이름 |
| 컴퓨터과학과 | 2018010873 | 전병규 |
| 컴퓨터과학과 | 2018010861 | 양소열 |
| 컴퓨터과학과 | 2018010877 | 정윤수 |
| 정보통계학과 | 2018010820 | 이지상 |

**목 차**

**1. 요약서**

**2. 서론**

2.1 추진 배경 및 목적

2.2 기존 현황 및 차별성

**3. 프로젝트 목표 및 주요내용**

3.1 프로젝트의 목표와 주요 내용

3.2 프로젝트 개발 내용

**4. 설계 및 구현**

**4.1 요구사항 분석**

4.1.1 기능 요구사항

4.1.2 비기능 요구사항

**4.2 시스템 설계**

4.2.1 개발 환경

**4.3 테스트 케이스 설계**

**4.4 구성요소별 기능 구현**

**4.4.1 Fronted**

4.4.1.1 UI설계

4.4.1.2 UI 개발

4.4.1.3 React 연동

**4.4.2 Back-End 구현**

4.4.2.1 User Service

4.4.2.2 Product & Cart Service

4.4.2.3 Order & Payment Service

4.4.2.4 Recommender System

**4.5 프로젝트 업무 분담**

**4.6 개인별 개발 일정**

**5. 개발 결과**

**6. 결론**

6.1 기대효과

6.2 활용 방안

**부록**

**1. 요약서**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **프로젝트 과목** | 종합설계 PBL | | | |
| **프로젝트명** | 트북이(AI 기반 노트북 추천 및 맞춤형 노트북 쇼핑몰) | | | |
| **팀 장** | **성 명** | 전병규 | **팀 명** | 브레인스토밍 |

**1.1 프로젝트 목표**

* 노트북을 구매할 때, 제품의 성능, 디자인, 가격, 브랜드 등 다수의 고려해야 할 요소가 존재한다. 대부분의 사람들은 이러한 요소를 고려하지 않고, 지인과 직원의 소개, 여러 커뮤니티와 유튜브에 의존하며 현명하게 소비할 권리를 보장받지 못하고 있다.
* 따라서 본 프로젝트는 사용자의 상황과 성향에 맞는 맞춤형 노트북을 AI를 통해 추천해 주고, 사용자의 활동을 분석하며, 미래 쇼핑 트렌드나 선호도를 예측하는 마케팅 전략을 보유하여 시장에서의 차별점을 두려 한다.

**1.2 프로젝트 내용**

* 사용자가 선호하는 노트북에 대한 정보를 수집한다.
* 사용자가 선호하는 노트북에 대한 데이터를 기반으로 개인에게 맞는 최적의 노트북을 추천해 준다.
* AI로 추천된 노트북뿐 아니라 이외로 자신이 사고 싶은 노트북을 직접 쇼핑몰에서 검색하고 결제할 수 있다.
* 상품의 정보를 AI 기술을 활용하여 자동으로 수집하고 분석하기 때문에 사용자가 노트북에 대한 정보를 추가적으로 조사할 필요가 없다. 그리고 최신 노트북 정보와 가격 변동에 대한 업데이트도 자동으로 이루어지기 때문에, 소비자 입장에서 신뢰할 수 있는 정보가 제공된다.

**1.3 프로젝트 개발에 따른 기대 성과**

* 사용자들은 보다 더 적합한 제품을 찾을 수 있으며, 보다 빠르고 간편하게 제품을 선택할 수 있게 되고, 쇼핑에 대한 만족도 상승
* AI 기술을 활용하여 상품 정보의 정확성이 향상되고, 사용자들은 보다 더 정확한 정보를 기반으로 제품 선택 가능
* 사용자들의 검색 이력 및 구매 이력 등을 분석하여 제품 추천뿐만 아니라, 미래의 쇼핑 트렌드나 사용자들의 선호도 등을 예측 가능
* AI 기술을 활용하여 다른 쇼핑몰과 차별화된 서비스를 제공할 수 있고, 이는 사용자의 이탈을 막고 유지 보수에도 도움.

**2. 서론**

**2.1 추진 배경 및 목적**

* 2021년 기준으로 전 세계 노트북 시장 규모는 약 2,000억 달러로 추산된다.
* 노트북 시장은 지속해서 성장하고 있고, COVID-19 팬데믹으로 인해 원격 근무 및 온라인 수업 등이 증가하면서 노트북 수요가 급증했다.
* 기술의 발전으로 노트북의 성능도 상향 평준화가 되어, 사용자가 선택할 수 있는 노트북의 폭이 넓다.
* 노트북은 한국인이 가장 많이 구매한 전자제품 1위로 꼽히는 제품으로 수요가 많다.
* 노트북을 구매할 때, 제품의 성능, 디자인, 가격, 브랜드 등 다수의 고려해야 할 요소가 존재한다.
* 대부분의 사람은 이러한 요소를 고려하지 않고, 지인과 직원의 소개, 여러 커뮤니티와 유튜브에 의존하며 현명하게 소비할 권리를 보장받지 못하고 있다.
* 메이저 3사의 100만원 상당의 노트북, 기타 해외업체의 50만원 상당의 노트북의 사양은 상이하지만, 소비자들은 메이저 3사의 정보만 알고 있어 편향된 소비를 보이는 경향이 있다.

**2.2 기존 현황 및 차별성**

**2.2.1 DANAWA**

텍스트, 스크린샷, 웹 페이지, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* DANAWA사이트의 상세 검색에서 브랜드, 가격, 사양 등을 선택하면 필터링을 거쳐 보여준다.
* 필터링을 거치더라도 노트북 제품이 너무 많아 자신에게 맞는 노트북을 선택하기 위해서는 추가적인 조사와 비교 검토가 필요하다.
* 노트북에는 Trade-Off 관계가 있어, 가격이 좋을수록 소비자에게 실질적으로 필요한 부품이나 기능들이 누락되어 있을 가능성이 있지만 DANAWA에서는 걸러주지 못한다.
* DANAWA에서 다양한 광고를 적극적으로 사용하고 있는데, 이러한 광고가 사용자 경험에 부정적인 영향을 미칠 가능성이 있다.
* 똑같은 노트북 이름에 여러 개의 다른 모델명들이 존재합니다. 예를 들어, LG그램을 사더라도 15ZD95Q-GX56K, 15ZD95Q76K 등 비슷해 보이는 여러 모델명이 존재해서 소비자들이 구매에 혼란이 있을 수 있다.
* 키워드로 검색할 때 DANAWA랑 스폰서를 제휴한 제품이 제일 상위권에 배치해 놓는 광고가 있어서, 소비자들의 현명한 소비를 힘들게 한다.

**2.2.2 유튜브/블로그 게시글**

**텍스트, 스크린샷, 웹 페이지, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

* 다수의 바이럴 광고가 존재한다.
* 특정 제품의 정보를 얻기 위해 유튜브나 블로그 글을 찾아보게 되면, 다수의 바이럴 마케팅 게시글이 존재한다.
* 실제 자기가 사용해 본 후기를 나타낸 게시글인지 광고 게시글인지 구분하기 어렵다.
* 일부 광고들은 자신의 이득을 위해 가성비가 떨어지는 노트북 제품들을 광고하는 경우도 있다.

**2.2.3 개선점 및 차별성**

* 기존 노트북 추천 사이트를 이용하게 되면 필터링을 하더라도 보여지는 제품이 많아서 자신에게 맞는 노트북을 선택하기 위해서는 추가적인 조사가 필요하게 된다는 문제점을 앞서 설명하였다. 하지만 ‘트북이’는 개인 맞춤형 AI 추천 서비스이기 때문에 자신의 현재 상황에 맞는 최적의 노트북을 추천해 준다. 예를 들어 사용자의 용도, 현재 예산에 따른 노트북 추천, 사용자가 선호하는 브랜드나 스펙에 따른 추천 등을 제공해 준다.
* 상품의 정보를 AI 기술을 활용하여 자동으로 수집하고 분석하기 때문에 사용자가 노트북에 대한 정보를 추가적으로 조사할 필요가 없다. 그리고 최신 노트북 정보와 가격 변동에 대한 업데이트도 자동으로 이루어지기 때문에 소비자 입장에서 신뢰 할 수 있는 정보가 제공된다.
* 다양한 쇼핑몰에서의 노트북의 가격을 비교해 주는 기능을 제공하여, 사용자가 구매 결정을 내릴 때 더 합리적인 선택을 할 수 있도록 돕는다.
* AI로 추천된 노트북 뿐 아니라 이외로 자신이 사고 싶은 노트북을 직접 쇼핑몰에서 검색하고 결제할 수 있다.

**3. 프로젝트 목표 및 주요내용**

**3.1 프로젝트 목표**

* 사용자의 상황과 성향에 맞는 맞춤형 노트북을 AI를 통해 추천해 주고, 사용자의 활동을 분석하며, 미래 쇼핑 트렌드나 선호도를 예측하는 마케팅 전략을 보유하여 시장에서의 차별점을 두려 한다.
* 사용자가 선호하는 노트북에 대한 정보를 수집한다.
* 사용자가 선호하는 노트북에 대한 데이터를 기반으로 개인에게 맞는 최적의 노트북을 추천한다.
* AI로 추천된 노트북뿐 아니라 이외로 자신이 사고 싶은 노트북을 직접 쇼핑몰에서 검색하고 결제가 가능하다.
* 상품의 정보를 AI 기술을 활용하여 자동으로 수집하고 분석하기 때문에 사용자가 노트북에 대한 정보를 추가로 조사할 필요가 없음. 그리고 최신 노트북 정보와 가격 변동에 대한 업데이트가 자동으로 이루어지기 때문에 소비자 입장에서 신뢰할 수 있는 정보가 제공된다.

**3.2 프로젝트 개발 내용**

텍스트, 도표, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

**4. 설계 및 구현**

**4.1 요구사항 분석**

**4.1.1 기능 요구사항**

|  |  |  |
| --- | --- | --- |
| 로그인 기능 | e-mail과 비밀번호를 입력하여 로그인하는 기능  세션 로그인도 구현하여 사용자가 원하는 로그인을 할 수 있도록 함. | |
| 입력 | e-mail, 비밀번호 |
| 출력 | 로그인 |
| 사용방식 선택 기능 | 사용자가 원하는 사용방식을 선택할 수 있는 기능 | |
| 입력 | AI 추천 모드, 일반 모드 |
| 출력 | AI 추천 모드와 일반모드 중 사용자가 선택한 방식 |
| 다중검색 필터 기능 | 원하는 필터를 설정하여 필터에 맞는 노트북을 제공 | |
| 입력 | 제조사, 용도, 무게, 가격대 등 원하는 필터 선택 |
| 출력 | 메인화면에 필터에 맞는 노트북 출력 |
| 노트북 상세정보  조회 기능 | 앱에서 제공하는 노트북에 대한 정보를 제공해 주는 기능. | |
| 입력 | 노트북 이미지 클릭 |
| 출력 | 해당 노트북의 상세정보 제공 |
| 장바구니 기능 | 사용자가 구매를 희망하는 노트북을 장바구니에 담을 수 있는 기능 | |
| 입력 | 노트북 상세정보 페이지 내 장바구니 담기 버튼 클릭 |
| 출력 | 장바구니 리스트에 추가 |
| 주문 기능 | 사용자가 원하는 노트북을 결제할 수 있도록 주문하는 기능 | |
| 입력 | 장바구니 페이지 내 구매하기 버튼, 상세정보 페이지 내 구매하기 버튼 |
| 출력 | 주문/결제 페이지로 이동 |
| 결제 기능 | 사용자가 원하는 노트북을 결제할 수 있도록 하는 기능 | |
| 입력 | 주문/결제 페이지 내 결제하기 버튼 |
| 출력 | 결제 페이지로 이동 |

**4.1.2 비기능 요구사항**

|  |  |  |
| --- | --- | --- |
| 성능 요구사항 | 응답시간 | 시스템은 사용자의 요청에 대해 2초 이내에 응답해야 한다. |
| 처리량 | 시스템은 초당 100개의 요청을 처리할 수 있어야 한다. |
| 확장성 | 시스템은 수평 및 수직 확장이 가능해야 하고, 트래픽 증가에 따라 선형적으로 확장될 수 있어야 한다. |
| 메모리  사용량 | 시스템은 효율적으로 메모리를 사용해야 한다. 메모리 누수나 비효율적인 자원 사용을 최소화해야 한다. |
| 로깅 및  모니터링 | 시스템은 로그를 적절히 기록하고 모니터링 해야한다. 성능 지표와 예외를 모니터링하여 시스템 성능을 지속적으로 개선할 수 있어야 한다. |
| 보안 요구사항 | 인증,  권한부여 | 시스템은 사용자를 인증하고 권한을 부여하여 민감한 데이터와 기능에 접근할 수 있는지 여부를 제어해야 한다. |
| 데이터  보호 | 시스템은 데이터의 기밀성, 무결성 및 가용성을 보호해야 한다. 적절한 데이터 암호화 및 액세스 제어 메커니즘을 사용해야 한다. |
| 가용성 요구사항 | 시스템  운영시간 | 시스템은 99.9%의 가용성을 제공해야 한다. 예정된 유지버스 시간을 제외한 경우에도 사용자가 접근할 수 있어야 한다. |
| 장애복구 | 시스템은 장애가 발생한 경우에도 빠른 복구를 제공해야 한다. 데이터 백업 및 복원, 장애 허용적 아키텍처를 고려해야 한다. |
| 유지보수성  요구사항 | 코드  가독성 | 시스템은 읽기 쉽고 이해하기 쉬운 코드로 작성해야 한다. 적절한 주석과 문서화를 통해 코드를 설명해야 한다. |
| 변경  용이성 | 시스템은 새로운 기능 추가나 기존 기능 수정에 대한 변경이 용이해야 한다. 높은 결합도와 낮은 의존성을 가진 모듈화된 아키텍처를 고려해야 한다. |
| 확장적 요구사항 | 수평확장 | 시스템은 증가하는 트래픽에 대해 서버를 수평으로 확장할 수 있어야 한다. 로드밸런싱, 분산 캐싱등을 활용해서 확장성을 개선해야 한다. |
| DB확장성 | 환시스템은 대량의 데이터를 처리하기 위해 데이터베이스를 확장할 수 있어야 한다. 샤딩, 파티셔닝등의 기술을 고려해야 한다. |
| 호환성 요구사항 | 플랫폼  호환성 | 시스템은 다양한 운영 체제, 브라우저, 디바이스에 호환되어야 한다. 크로스 플랫폼 테스트와 호환성 테스트를 수행해야 한다. |
| API  호환성 | 시스템은 변경이나 업그레이드로 인해 기존 API를 손상시키지 않고 호환성을 유지해야한다. |
| 규정준수  요구사항 | 암호화  요구사항 | 시스템은 민감한 데이터를 암호화하여 저장하고 전송해야 한다. 암호화 알고리즘과 키 관리를 포함한 보안 메커니즘을 사용해야 한다. |
| 사생활  보호 | 시스템은 사용자의 개인정보와 사적인 데이터를 적절히 보호해야 한다. 개인정보 처리 방침, 동의 절차 및 사용자 데이터 액세스 제어등을 포함해 개인정보 보호에 대한 규정을 준수해야 한다. |

**4.2 시스템 설계**

**4.2.1 개발 환경**

|  |  |  |
| --- | --- | --- |
| SW | 개발 언어 | CSS, JavaScript, Java, Python |
| 편집기 | VS Code, IntelliJ IDEA, PyCharm |
| DBMS | MySQL |
| 서버 | AWS Elastic Beanstalk, S3, CloudFront |
| 디자인 | Figma |
| 협업 | GitHub, 구글 문서, Notion |
| Front-End | HTML, CSS, Javascript, React |
| Back-End | Spring Boot, Hibernate, FastAPI |
| DevOps | Docker, Github Action |
| AI | Kaggle, Tensorflow, Pytorch, Doc2Vec |
| 결제 API | Iamport API |
| 구동 환경 | OS | MacOS, WIndow |

**4.3 테스트케이스 설계**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **대분류** | **중분류** | **소분류** | **내용** | **결과** | **비고** |
| 1 | 초기페이지 | 로그인 | 기본 로그인 | 사용자가 유효한 아이디와 비밀번호로 기본 로그인을 시도합니다. | **P** | 소셜로그인 기능 (카카오톡, 구글) |
| 2 | 카카오톡 로그인 | 사용자가 카카오톡 계정으로 로그인을 시도합니다. | **P** |  |
| 3 | 구글 로그인 | 사용자가 구글 계정으로 로그인을 시도합니다. | **P** |  |
| 4 | 비밀번호 찾기 | 비밀번호 재설정 요구조건 | 사용자가 비밀번호 재설정을 요청합니다. | **P** | 회원가입 단계에서 기입한 이메일과 전화번호를 입력해야 함. |
| 5 | 비밀번호 재설정 | 사용자가 새 비밀번호를 입력하고 변경을 시도합니다. | **P** | 비밀번호, 비밀번호 확인 데이터가 일치해야 함. |
| 6 | 회원가입 | 회원가입 정보 기입 | 사용자가 필요한 회원가입 정보를 입력합니다. | **P** | 이메일, 비밀번호, 비밀번호 확인, 이름, 주소, 전화번호 기입 |
| 7 | 사용방식 선택 | 사용자가 회원가입 과정에서 서비스 사용 방식을 선택합니다. | **P** | AI추천 모드, 일반 모드 선택 가능 |
| 8 | 메인페이지 | 필터링 | 제조사 | 사용자가 제조사를 선택하여 결과를 필터링합니다. | **P** | SAMSUNG, LG, RAZER, ASUS, MSI, 레노버 |
| 9 | 용도 | 사용자가 노트북의 용도(예: 게임용, 사무용 등)를 선택하여 결과를 필터링합니다. | **P** | 고사양게임, 문서작업, 그래픽작업, 가성비 최고, 트부기 추천 |
| 10 | 무게 | 사용자가 노트북의 무게(예: 가벼운, 중간, 무거운 등)를 선택하여 결과를 필터링합니다. | **P** | ~1kg, ~1.5kg, ~2kg, ~3kg |
| 11 | 가격대 | 사용자가 가격대를 선택하여 결과를 필터링합니다. | **P** | ~40만원, 40~70만원, 70~100만원, 100~200만원, 200만원 이상 |
| 12 | 목록 | 사용자가 필터링된 노트북 목록을 확인합니다. | **P** | 무한 스크롤 구현 |
| 13 | 노트북 상세 | 장바구니 담기 | 사용자가 특정 노트북의 상세 페이지로 이동하고 장바구니에 담습니다. | **P** |  |
| 14 | 구매하기 | 사용자가 장바구니에 담긴 노트북을 구매합니다. | **P** |  |
| 15 | 주문 페이지 | 장바구니 | 물품 선택 | 사용자가 장바구니에서 구매할 물품을 선택합니다. | **P** | 체크 버튼을 통해 선택 가능 |
| 16 | 구매하기 | 사용자가 장바구니에 있는 물품을 구매합니다. | **P** |  |
| 17 | 주문/결제 | 결제수단 | 사용자가 주문/결제 과정에서 결제수단을 선택합니다. | **P** | 카카오페이 결제, 일반 결제 가능 |
| 18 | 결제하기 | 사용자가 선택한 결제수단으로 결제를 진행합니다. | **P** |  |
| 19 | 주문 목록 | 배송 조회 이동 | 사용자가 주문 목록에서 특정 주문의 배송 조회 페이지로 이동합니다. | **P** |  |
| 20 | 주문 목록 표시 | 사용자가 주문 목록을 확인합니다. | **P** | 지금까지 주문한 모든 제품 목록을 보여줌 |
| 21 | 배송 조회 | 배송 상태 표시 | 사용자가 배송 조회에서 특정 주문의 배송 상태를 확인합니다. | **P** | 배송 준비 중, 배송 중, 배송 완료 정보 표시 (이미지 형태) |
| 22 | 마이페이지 | 로그아웃 | 사용자가 마이페이지에서 로그아웃합니다. | **P** |  |
| 23 | 주소 설정 | 사용자가 마이페이지에서 주소를 설정합니다. | **P** |  |
| 24 | 주문 목록 | 사용자가 마이페이지에서 주문 목록을 확인합니다. | **P** |  |
| 25 | 공지사항 | 사용자가 마이페이지에서 공지사항을 확인합니다. | **P** |  |
| 26 | 고객센터 | 사용자가 마이페이지에서 고객센터에 접근합니다. | **P** |  |
| 27 | 환경설정 | 사용자가 마이페이지에서 환경설정을 변경합니다. | **P** |  |
| 28 | 비밀번호 변경 | 사용자가 마이페이지에서 비밀번호를 변경합니다. | **P** |  |
| 29 | 회원 탈퇴 | 사용자가 마이페이지에서 회원 탈퇴를 진행합니다. | **P** |  |
| 30 | 버전 정보 | 사용자가 마이페이지에서 현재 버전 정보를 확인합니다. | **P** |  |
| 31 | 추천페이지 | 추천 화면 | 슬라이드 페이지 형태 추천페이지 | 사용자가 추천페이지에서 슬라이드 형태의 추천 노트북 목록을 확인합니다. | **P** | 추천 결과 10개 제공 (유사도를 사용자가 이해하기 쉽게 정확도로 변경하여 표기) |

**4.4 구성요소별 기능 구현**

**4.4.1 Front-End 구현**

**4.4.1.1 UI 설계스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명**

저희 웹 · 앱은 로그인, 회원가입, 사용방식 선택, 메인, 노트북 상세, 노트북 추천, 장바구니, 주문, 마이 페이지, 주문 목록, 배송 조회 등의 다양한 화면으로 구성되어 있다. 특히, 노트북 추천 기능은 주요 기능으로서 사용자가 직관적으로 원하는 노트북의 세부 사항을 선택할 수 있도록 설계되었다.

노트북 추천 화면은 슬라이드 형태로 구성되어 있어 한 번의 추천마다 다양한 노트북 선택지를 제공한다. 각 추천마다 5개의 상품을 보여주어 사용자가 다양한 옵션을 살펴볼 수 있도록 하였다. 이를 통해 사용자는 편리하게 원하는 노트북을 선택할 수 있다.

로그인 기능은 다양한 형식을 지원하기 위해 소셜 로그인 기능도 포함되어 있다. 사용자는 웹에 로그인할 때마다 AI 추천 모드와 일반 모드 중 선택할 수 있다. 이를 통해 사용자는 자신의 취향과 선호도에 따라 AI 추천을 받을 수도 있고, 직접 노트북을 탐색할 수도 있다.

또한, 노트북 추천과 함께 사용자가 실제로 구매할 수 있도록 구매 및 결제, 그리고 배송 기능을 담당하는 화면도 설계되었다. 사용자는 노트북 추천을 받은 후에 원하는 상품을 선택하고 구매할 수 있다. 결제와 배송 과정은 간편하고 신속하게 처리되어 사용자가 원하는 노트북을 빠르게 소유할 수 있도록 한다.

이렇게 다양한 화면과 기능을 통해 사용자는 편리하게 로그인하고 웹 · 앱을 이용할 수 있으며, 노트북 추천과 구매 과정을 원활하게 경험할 수 있다. 본 웹 · 앱은 사용자에게 멋진 디자인과 직관적인 기능을 제공하여 노트북 구매와 사용에 대한 최상의 경험을 제공한다.

**4.4.1.2 UI 개발텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명**

HTML, CSS, JS를 활용하여 웹 · 앱 형태로 개발되었다. 이 웹 · 앱은 사용자가 데스크탑과 모바일 기기에서 일관된 UI를 경험할 수 있도록 설계되었다. 팀원들은 효율적인 개발을 위해 각자 담당한 부분을 섹션으로 나누어 작업하였다. 사용자, 주문, 결제, 제품, 장바구니 등과 같은 각 섹션은 개별적으로 화면을 개발하였다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

특히, AI 필터링 화면은 사용자의 선택에 따라 진행 사항이 업데이트되는 Progress 바를 갖추었다. 이 기능을 통해 사용자가 선택을 할 때마다 진행 상황을 시각적으로 확인할 수 있다. 사용자가 필터를 선택하고 설정할 때마다 Progress 바가 업데이트되어 진행 과정을 보다 효율적으로 파악할 수 있게 되었다.

이렇게 웹 · 앱을 개발함으로써 사용자는 데스크탑과 모바일 기기에서 일관된 UI를 경험할 수 있으며, AI 필터링 기능을 사용할 때도 진행 상황을 시각적으로 확인할 수 있다. 멋진 디자인과 함께 사용자에게 좋은 경험을 제공하는 웹 · 앱이 되도록 노력하고 있다.

**4.4.1.3 React 연동**

**스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

기존에 HTML, CSS, JS로 개발된 페이지를 React 컴포넌트로 재구성하였다. 이러한 전환을 통해 사용자에게 더욱 부드러운 동적 페이지를 제공할 수 있으며, SPA(Single Page Application) 구조를 통해 웹 앱에 최적화된 사용자 경험을 제공한다.

React 컴포넌트를 사용하면 각 부분을 독립적인 모듈로 분리하여 관리할 수 있다. 이를 통해 코드의 재사용성을 높이고 개발 과정을 효율화 할 수 있다. 또한, React의 가상 돔(Virtual DOM)을 활용하여 효율적인 렌더링을 가능하게 하여 성능을 향상시킬 수 있다.

추가적으로, Axios와의 연동을 통해 프론트엔드와 백엔드 간의 통신을 처리하였다. Axios는 HTTP 클라이언트 라이브러리로, 비동기적인 방식으로 서버와의 데이터 교환을 처리할 수 있다. 이를 통해 사용자의 요청에 따라 서버와 실시간으로 데이터를 주고받을 수 있고, 웹 · 앱의 기능을 확장하고 향상시킬 수 있다.

이렇게 React 컴포넌트로의 전환과 Axios와의 연동을 통해 더욱 부드럽고 동적인 페이지를 제공하며, 프론트엔드와 백엔드 간의 데이터 통신을 원활하게 처리하는 웹 · 앱을 구축하였다. 사용자에게 탁월한 경험과 기능을 제공하는 웹 · 앱을 개발하기 위해 노력하고 있다.

**4.4.2 Back-End 구현**

**4.4.2.1 User Service**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**회원인증**

저희 서비스는 사용자들의 개인정보 보호와 안전한 인증을 최우선으로 여기고 있다. 이를 위해 스프링 부트에서 제공하는 BCPasswordEncrypt Bean을 활용하여 사용자의 패스워드를 안전하게 암호화하여 데이터베이스에 저장하고 있다. 이를 통해 사용자들의 패스워드가 해시화되어 저장되어 악의적인 공격으로부터 안전하게 보호된다.

뿐만 아니라, 우리는 사용자들이 편리하게 서비스를 이용할 수 있도록 OAuth 2.0 라이브러리를 활용하여 구글, 카카오, 네이버와 같은 소셜 로그인 기능을 구현하였다. 이를 통해 사용자들은 기존에 생성한 소셜 미디어 계정을 활용하여 간편하게 로그인하고 우리 서비스를 이용할 수 있다.

또한, 우리는 Spring Security의 FilterChain을 이용하여 기존에 사용하던 Basic 인증 방식을 비활성화하고, 보다 안전하고 효율적인 JWT 토큰을 활용한 인증 방식을 구현하였다. 사용자가 로그인에 성공하면, 서버에서는 JWT 토큰을 발급하여 사용자에게 전달한다. 이 JWT 토큰은 Spring SecurityPersistContext에 안전하게 저장되어 서버 측에서 사용자의 로그인 정보를 관리한다.

우리의 작업은 보안과 사용자 경험을 모두 고려하여 구성되었다. 사용자들은 안전하게 개인정보를 보호받으며, 편리하고 간편한 소셜 로그인 기능을 통해 우리의 서비스를 쉽게 이용할 수 있다. 앞으로도 저희는 보다 발전된 기술과 최상의 보안을 제공하기 위해 노력할 것이다.

**RestAPI 변환**

저희는 최신 웹 애플리케이션의 요구에 맞춰 서버 측 렌더링 (Server Side Rendering, SSR)에서 클라이언트 측 렌더링 (Client Side Rendering, CSR)으로의 전환 작업을 진행하였다. 이를 위해 OAuth 2.0 라이브러리를 사용하지 않고, 직접 RestTemplate을 활용하여 각 리소스 서버로부터 데이터를 가져오는 REST API를 구축하였다.

우선, 프론트엔드에서는 OAuth 리소스 서버에서 발급된 임시 코드를 사용하여 백엔드에서 Access Code를 받을 수 있도록 API를 호출한다. 이 Access Code를 통해 백엔드에서는 정상적인 로그인 절차를 수행하고, 소셜 로그인한 회원 정보를 데이터베이스에 저장한다. 로그인이 완료되면, 백엔드는 Response Header에 Authorization에 Bearer JWT 토큰을 추가하여 반환한다.

프론트엔드는 받은 JWT 토큰을 쿠키에 저장하여 보관하며, 회원 정보가 필요한 API를 호출할 때 해당 JWT 토큰을 함께 전송한다. 이를 통해 프론트엔드는 백엔드에서 인증 작업을 수행할 수 있다. JWT 토큰은 사용자의 인증을 나타내며, 보안 및 편의성을 제공하여 사용자가 손쉽게 서비스를 이용할 수 있도록 한다.

또한, 우리는 네이버, 카카오, 구글 등 인기 있는 3개의 소셜 로그인 서비스와 연동하여 사용자들이 우리 서비스를 더욱 쉽게 이용할 수 있도록 구성하였다. 이를 통해 사용자들은 자신의 소셜 로그인 정보를 활용하여 간편하게 회원 가입하고 로그인할 수 있다.

저희의 작업은 최신 웹 개발 표준에 부합하며, 보안과 효율성을 고려하여 사용자 경험을 향상시키는 데 집중하였다. 앞으로도 더욱 발전된 기술과 편의성을 제공하기 위해 노력할 것이다.

이렇게 암호화된 패스워드 저장, OAuth 2.0 소셜 로그인, JWT 토큰을 이용한 인증, Spring Security FilterChain을 활용한 보안 기능 등을 구현함으로써 우리 서비스는 사용자들에게 안전하고 편리한 인증 및 로그인 경험을 제공한다.

**4.4.2.2 Product & Cart Service**

**텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명**

**요구사항 분석 및 환경구축**

저희 서비스를 구현하기 위해 체계적인 도메인 설계와 엔티티 설계를 수행하였다. 도메인 설계를 통해 관계를 명확히 파악할 수 있으며, 각 기능의 엔티티 설계를 통해 필요한 정보를 효율적으로 관리하고 활용할 수 있게 되었다. 이를 바탕으로 API 명세서를 작성하여 개발 시 효율성을 극대화하였다.

뿐만 아니라 저희는 마이크로 서비스 아키텍처를 통해 각 기능을 독립적으로 개발하고 연동하였다. 마이크로 서비스 아키텍처를 채택함으로써 각 기능 간의 결합도를 낮추고 유연한 개발과 확장성을 도모할 수 있다. 또한, 마이크로 서비스 간의 통신을 위해 표준화된 API를 사용하여 각 기능을 효과적으로 연결하였다.

**상품 기능 개발**

상품 마이크로 서비스는 사용자들에게 편리하고 정확한 상품 검색 경험을 제공하기 위해 다양한 기능을 제공합니다. 전체 상품 조회 기능을 통해 사용자들은 모든 상품을 한 눈에 확인할 수 있다. 이를 통해 다양한 상품 정보를 탐색하고 비교할 수 있다.

또한, 다중 필터 검색 기능은 사용자들이 원하는 조건에 따라 상품을 세밀하게 검색할 수 있도록 해준다. 가격, 제조사, 무게 등 다양한 필터를 조합하여 사용자들은 원하는 상품을 빠르고 정확하게 찾을 수 있다. 이는 사용자들의 개인적인 선호도나 요구사항에 맞춤화 된 상품 검색 경험을 제공하는데 큰 도움이 된다.

**장바구니 기능 개발**

장바구니 마이크로 서비스는 사용자들이 손쉽게 상품을 관리하고 구매 과정을 원활하게 진행할 수 있도록 다양한 기능을 제공한다. 이 서비스는 장바구니 상품 조회, 장바구니 상품 삭제, 장바구니 상품 수량 증가/감소 등과 같은 기능을 포함하고 있다.

장바구니 상품 조회 기능은 사용자들이 현재 장바구니에 담긴 상품 목록을 확인할 수 있도록 해준다. 사용자는 한눈에 자신이 선택한 상품들을 확인하고, 각 상품의 가격과 수량 등의 정보를 확인할 수 있다. 이를 통해 사용자들은 구매하고자 하는 상품들을 장바구니에 편리하게 모아둘 수 있다.

또한, 장바구니 상품 삭제 기능은 사용자들이 장바구니에 담긴 특정 상품을 제거할 수 있도록 해준다. 사용자는 더 이상 구매하지 않을 상품이나 잘못 추가한 상품을 선택하여 삭제할 수 있다. 이를 통해 사용자들은 장바구니를 정리하고 필요한 상품들만 남기는데 도움을 받을 수 있다.

또한, 장바구니 상품 수량 증가/감소 기능은 사용자들이 장바구니에 담긴 특정 상품의 수량을 조정할 수 있도록 해준다. 사용자는 원하는 상품의 수량을 증가시키거나 감소시킬 수 있으며, 이를 통해 자신의 구매 계획에 맞춰 상품 수량을 조절할 수 있다. 이는 사용자들이 유연하게 상품 수량을 관리하고 구매량을 조정할 수 있는 장점을 제공한다.

이처럼 장바구니 마이크로 서비스는 사용자들에게 편의성과 유연성을 제공하기 위해 이러한 여러 기능을 제공한다. 사용자들은 상품 조회, 삭제, 수량 조정 등의 기능을 통해 자신의 장바구니를 효율적으로 관리할 수 있으며, 구매 과정을 원활하게 진행할 수 있다. 이는 사용자들에게 편리한 쇼핑 경험을 제공하는데 큰 역할을 한다.

**양식의 맨 위**

**양식의 맨 아래**

**4.4.2.3 Order & Payment Service**

텍스트, 스크린샷, 소프트웨어, 디자인이(가) 표시된 사진

자동 생성된 설명

**주문 기능 개발**

사용자가 제품을 간편하게 주문할 수 있도록 개선된 주문 기능을 제공하는 것이 주문 마이크로 서비스의 주요 목표였다. 이를 위해, 우리는 체계적으로 주문 정보를 관리할 수 있는 주문 엔티티를 설계하였고, 이를 기반으로 주문 API를 개발하였다. 주문 API는 사용자가 제품을 선택하고 주문하는 데 필요한 모든 기능을 제공하며, 주문이 접수되면 제품 재고를 실시간으로 업데이트하고 잔액을 차감하는 기능을 포함하고 있다. 이를 통해, 시스템은 주문 처리와 재고 관리를 신속하게 수행하고, 사용자에게는 원활한 주문 경험을 제공할 수 있다.

이러한 주문 기능은 MSA 환경 내에서 독립적인 마이크로 서비스로 운영되며, 외부와의 모든 통신은 API Gateway를 통해 이루어진다. 이렇게 별도의 마이크로 서비스로 주문 기능을 분리하면서, 시스템의 효율성과 유연성이 크게 향상되었다. 또한, 각 마이크로 서비스는 독립적으로 확장 가능하여, 주문량이 증가하더라도 시스템의 성능에 영향을 미치지 않는다.

데이터 동기화를 위해, 우리는 강력하고 빠른 Kafka를 통합하였다. Kafka는 ZooKeeper 서버, Confluent 서버, 그리고 Kafka 서버 등을 통해 빠르고 안정적인 데이터 전달을 보장한다. 이렇게 주문 마이크로 서비스와 Kafka 사이의 통합을 통해, 주문 처리에 관련된 데이터는 신속하게 동기화되어, 사용자에게는 항상 최신의 정보가 제공될 수 있다. 이러한 구조는 높은 처리량과 확장성을 보장하며, 신뢰성 있는 유연한 데이터 흐름을 유지하게 하도록 구현하였다.

또한, Spring Eureka를 주 서버로 도입하여 시스템의 신뢰성과 확장성을 더욱 높였다. 이는 마이크로 서비스들 간의 효율적인 통신을 가능하게 하며, 시스템의 안정적인 운영을 지원한다. 최종적으로, 이 모든 구성 요소들이 결합되어, 우리의 MSA 환경이 사용자에게 안정적이고 빠르게 반응하는 주문 기능을 제공하는 데 중요한 역할을 담당하고 있다. 이렇게 하여, 사용자는 제품을 편리하게 주문하고, 그에 따른 정보를 신속하게 받을 수 있으며, 이는 사용자의 만족도를 높이는 데 큰 도움이 된다.

**결제 엔티티 설계**

사용자가 제품을 간편하게 주문하고 안정적으로 결제할 수 있도록 하는 것이 우리의 주요 목표였다. 이를 위해, 현재 체계적으로 주문 정보를 관리할 수 있는 주문 엔티티를 설계하였다.

제공될 주문 API는 사용자가 제품을 선택하고 주문하는 데 필요한 모든 기능을 제공하며, 주문이 접수되면 제품 재고를 실시간으로 업데이트하고 잔액을 차감하는 기능을 포함하고 있다.

결제 API는 I'mport 결제 서비스를 활용하여, 결제 완료 시 사용자에게 안정적인 결제 확인을 제공할 것으로 계획 중에 있다. 이를 통해, 시스템은 주문 처리와 재고 관리, 그리고 결제 처리를 신속하게 수행하고, 사용자에게는 원활한 주문 및 결제 경험을 제공할 수 있다.

이러한 주문 및 결제 기능은 MSA 환경 내에서 독립적인 마이크로 서비스로 운영되며, 외부와의 모든 통신은 API Gateway를 통해 이루어진다. 이렇게 별도의 마이크로 서비스로 주문 및 결제 기능을 분리하면서, 시스템의 효율성과 유연성이 크게 향상되었다. 또한, 각 마이크로 서비스는 설계 상, 독립적으로 확장 가능하여 주문량이 증가하더라도 시스템의 성능에 영향을 미치지 않는다.

**4.4.2.4 Recommender System**

**텍스트, 스크린샷, 폰트, 웹 페이지이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

다나와 사이트에서 노트북 데이터를 수집하기 위해 웹 스크래핑 자동화를 구현했다. 이를 위해 Selenium과 BeautifulSoup를 사용했다. 또한, FastAPI를 활용하여 24시간마다 스크래핑을 시도하여 최신 정보 기반 서비스를 제공한다.

우선, Selenium은 웹 브라우저를 자동으로 제어할 수 있는 도구이다. 다나와 사이트에 접속하고 상호작용하기 위해 Selenium을 사용했다. Selenium을 통해 브라우저를 제어하고 필요한 정보를 추출하기 위해 BeautifulSoup를 사용했다. BeautifulSoup은 HTML 및 XML 문서를 파싱하고 원하는 데이터를 추출하는 파이썬 라이브러리이다.

다나와 사이트의 노트북 페이지에 접속하여 원하는 정보를 수집하는 코드를 구현했다. Selenium을 사용하여 사이트에 접속하고 필요한 정보를 찾기 위해 웹 요소를 탐색했다. 그런 다음, BeautifulSoup를 사용하여 필요한 데이터를 추출했다. 이러한 과정을 반복하여 여러 노트북의 정보를 수집했다.

데이터 수집이 완료된 후에는 FastAPI를 사용하여 서비스를 구현했다. FastAPI는 빠르고 현대적인 웹 프레임워크로, 간단한 API 엔드포인트를 쉽게 작성할 수 있다. 24시간마다 스크래핑을 시도하기 위해 스케줄링 작업을 설정했다. 이를 통해 정해진 간격으로 데이터를 업데이트하고 최신 정보를 서비스할 수 있었다.

이렇게 구현한 코드는 지속적으로 실행되며, 다나와 사이트의 노트북 데이터를 실시간으로 수집하여 최신 정보를 기반으로 서비스를 제공한다.

텍스트, 스크린샷, 로고, 디자인이(가) 표시된 사진

자동 생성된 설명

초기 Cold Start 문제를 해결하기 위해 군집화 및 유사도 기반 추천 방식을 채택하여 추천 시스템을 구현하고자 한다. 이를 위해 사용자가 제공 서비스에서 특정 노트북을 검색하거나 볼 경우, 그와 유사한 스펙과 성능을 가진 노트북을 추천해주는 방식을 사용했다. 이를 통해 초기 추천 시스템을 구축하고, 그 후에 사용자 피드백을 수집하여 AI 추천 서비스의 성능을 향상시킬 것이다.

사용자가 검색하거나 본 노트북에 대한 정보를 수집합니다. 이를 기반으로 각 노트북의 스펙, 성능 등을 벡터화 한다.

벡터화 된 데이터를 활용하여 유사도 측정을 통해 비슷한 특성을 가진 노트북들을 추천한다.

그 후 사용자의 피드백을 수집하기 위해 추천된 노트북들에 대한 평가나 선호도를 기록한다. 예를 들어, 사용자가 특정 노트북을 선택하거나 구매하는 경우, 해당 정보를 수집한다.

사용자의 피드백을 이용하여 추천 시스템을 개선한다. 사용자가 선호하는 노트북 특성이나 성능을 파악하고, 이를 반영하여 추천 알고리즘을 업데이트 한다.

위의 방법을 통해 초기 추천 시스템을 구현하고 사용자 피드백을 수집함으로써, AI 추천 서비스의 성능을 지속적으로 향상시킬 수 있다. 또한, 사용자의 취향과 요구사항에 더욱 맞는 노트북을 추천하여 만족도를 높이고, Cold Start 문제를 극복할 수 있다.

**텍스트, 스크린샷, 폰트, 원이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명**

[노트북 스펙 데이터 전처리]

우선 수집한 노트북 스펙 데이터를 하나의 텍스트 문서로 결합한다. 예를 들어, 노트북의 브랜드, 모델명, 프로세서, 메모리, 디스플레이 크기, 저장 용량 등의 정보를 하나의 문장으로 구성한다.

그 후 문장의 텍스트를 전처리 과정을 거쳐 토큰화하고, 불용어(stop words)나 특수 문자 등을 제거하여 텍스트 데이터를 정제했다.

[벡터 표현]

정제된 텍스트 데이터를 토큰화하여 임베딩 기법(Doc2Vec)을 사용하여 벡터로 표현한다.

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명**

[유사도 측정:]

코사인 유사도를 활용하여 벡터 간의 유사성을 측정한다. 코사인 유사도는 벡터 간 각도의 코사인 값을 계산하여 유사성을 측정하는 방법이다.

노트북 스펙 데이터를 벡터로 표현한 후, 사용자가 검색하거나 본 노트북과 다른 노트북들 간의 코사인 유사도를 계산한다.

코사인 유사도 값이 높은 순으로 다른 노트북을 추천 목록에 추가한다. 이를 통해 사용자가 관심을 가질만한 유사한 노트북을 추천한다.

위의 방법을 통해 수집한 노트북 스펙 데이터를 자연어 처리(NLP)를 통해 문장으로 취급하고, 코사인 유사도를 활용하여 유사성을 측정하는 추천 시스템을 구현했다.

이를 통해 더 정확하고 개인화된 추천을 제공하며, Cold Start 문제를 극복하고 사용자 경험을 향상시킬 수 있다.

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명**

[AI 기반 추천시스템 - AI서버 연동을 통한 추천 페이지 개발]

**텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명**

요약하자면, ‘트북이’는 기존의 단순 라벨링 군집화 방식과 다르게 노트북 수집 데이터를 활용한 추천 시스템에서 자연어 처리(NLP)를 적용한다. 이를 위해 수집한 노트북 스펙 데이터를 하나로 합쳐 문장으로 취급하고, 이를 NLP 기법을 활용하여 유사도를 측정하는 방식을 사용하고 있다.

(코사인 유사도를 기반으로 유사성을 측정하고 있다.)

**4.5 프로젝트 업무 분담**

|  |  |  |
| --- | --- | --- |
| 팀 구성 | | 역할 및 책임 |
| 전 병 규 | 팀 장 | · 도메인, Product엔티티 설계  · API 명세서 작성  · Product UI구현  · Product 마이크로 서비스 API 구현  · Cart 마이크로 서비스 API 구현 |
| 양 소 열 | 팀 원 | · User 엔티티 설계  · Spring Security를 통한 인증과 인가 구현  · 카카오, 구글, 네이버 소셜 로그인 구현  · JWT 토큰 발급 기능 구현  · User 마이크로 서비스 API 구현 · 리액트와 백엔드간 인증과 인가 로직 연동  · FastAPI를 통한 실시간 크롤링 서버 구현  · FastAPI를 통한 추천 알고리즘 연동 |
| 정 윤 수 | 팀 원 | · 주문 엔티티 설계  · 주문 마이크로 서비스 API 구현  · 결제 마이크로 서비스 API 구현  · React 연동  · I’mport API 연동  · Kafka 연동 |
| 이 지 상 | 팀 원 | · 필터링 화면구상  · 노트북 데이터 크롤링 및 스크래핑  · 데이터 전처리  · AI추천알고리즘 구현 및 모델 성능 평가 |

**4.6 개인별 개발 일정**

|  |  |  |
| --- | --- | --- |
| 종합설계PBL 프로젝트 중간 수행 일정 | | |
|  | 전병규 | 양소열 |
| 5주차  3/27~ | · IntelliJ, Visual studio code, Spring boot 개발 환경 구축  · MSA 아키텍쳐 학습  · JPA 학습 | · IntelliJ, Visual studio code, Spring boot 개발 환경 구축  · MSA 아키텍쳐 학습  · JPA 학습 |
| 6주차  4/3~ | · MSA 아키텍쳐 학습  · 전체 도메인 설계 | · Spring Security 필터와 동작원리 이해  · Spring Security 커스텀 필터 만들어보기(70%) |
| 7주차  4/10~ | · Product 엔티티 설계  · API 명세 작성  · UI 및 기능 설계 | · 기존 Basic Auth 방식의 시스템의 이해  · Session 방식의 로그인 이해 및 Session 방식 로그인 구현 |
| 8주차  4/17~ | · UI 및 기능 설계  · Spring Eureka server 연결  · API Gateway 연결  · Product 마이크로 서비스 구현(10%) | · Session 로그인의 단점을 보완한 JWT방식 학습  · JWT 인증, 인가 필터 구현  · 회원가입 페이지 제작(React) |
| 9주차  4/24~ | · UI 개발(상품 상세 정보 페이지, 30%) | · OAuth2.0 로그인 방식 학습  · 네이버, 카카오, 구글 로그인 구현  · 리액트와 구글 로그인 연동 테스트(React) |
| 10주차  5/1~ | · UI 개발(상품 상세 정보 페이지, 100%)  · Product 마이크로 서비스 구현 30% | · RestAPI 연동을 위해 OAuth 로그인 로직 수정  · 직접 리소스 API 호출을 통한 액세스 토큰 획득 |
| 11주차  5/8~ | · UI 개발(메인 화면 페이지, 70%)  · Product 마이크로 서비스 구현 70% | · OAuth 로그인 방식에 JWT 토큰 반환 로직 작성  · JWT 토큰 발급 테스트 작성 (30%) |
| 12주차  5/15~ | · UI 개발(메인 화면 페이지, 100%)  · React 변환(10%)  · Kafka 개발 환경 구축 | · MSA 아키텍처에서 인증, 인가 처리 로직구현(10%)  · JWT 토큰 발급 테스트 작성 (100%) |
| 13주차  5/22~ | · Product 마이크로 서비스 구현 100%  · 동적쿼리 작성을 위한 Querydsl 학습  · 다중 검색 필터 api 구현 100% | · MSA 아키텍처에서 인증, 인가 처리 로직구현(100%)  · React변환 및 변환으로 인한 변경된 CSS 속성 수정  · FastAPI 스케쥴링 및 ORM 학습 |
| 14주차  5/29~ | · Cart 마이크로 서비스 구현 100%  · Product 마이크로 서비스 연동 | · 추천, 실시간 크롤링을 위한 FastAPI 서버 환경구축  · 스케쥴링을 통한 실시간 크롤링 구현 |
| 15주차  6/5~ | · Cart 마이크로 서비스 연동  · Doker를 통한 백엔드 배포 | · FastAPI 서버에 추천 알고리즘 연동을 통한 노트북 추천 API 추가  · React와 아임포트 API 연동 후 카카오페이, KCP PG 서비스 추가  · React와 백엔드 API 연동 및 프론트 배포 |

|  |  |  |
| --- | --- | --- |
| 종합설계PBL 프로젝트 중간 수행 일정 | | |
|  | 정윤수 | 이지상 |
| 5주차  3/27~ | · IntelliJ, Visual studio code, Spring boot 개발 환경 구축  · MSA 아키텍쳐 학습  · JPA 학습 | · 추천 알고리즘에 대한 기본 개념과 원리에 대해 학습  · 사용자 기반(User-based) 추천과 아이템 기반(Item-based) 추천에 대한 차이와 장단점 분석 |
| 6주차  4/3~ | · 임의 회원 마이크로 서비스 구현  · 임의 상품 마이크로 서비스 구현  · Spring Eureka, API Gateway 연결 | · Collaborative Filtering, Content-Based Filtering, Hybrid Recommender 등 추천 알고리즘의 다양한 유형 탐색. |
| 7주차  4/10~ | · UI 및 기능 설계(30%): 장바구니, 주문, 마이 페이지  · 주문 마이크로 서비스 구현(10%) | · 다나와 사이트의 구조와 데이터 패턴을 분석하기 위해 다나와 사이트의 HTML 구조 분석.  · 다나와 사이트에서 제품 목록을 가져오기 위한 URL 패턴과 페이지 구조 파악.  · 제품 이름, 가격, 상세 스펙 등 필요한 데이터  추출하기 위한 HTML 태그와 CSS 선택자 분석. |
| 8주차  4/17~ | · UI 및 기능 설계(100%): 장바구니, 주문, 마이 페이지  · 주문 마이크로 서비스 구현(30%) | · 다나와 사이트에서 필요한 데이터를 추출하기 위한 크롤러 구현.  · Python의 requests 라이브러리를 사용하여 다나와 사이트에 HTTP 요청.  · BeautifulSoup 라이브러리를 사용하여 HTML 페이지를 파싱, 필요한 데이터를 추출하는 코드 작성. |
| 9주차  4/24~ | · UI 개발(30%): 장바구니, 주문, 마이 페이지  · 주문 마이크로 서비스 구현(50%) | · 데이터의 정확성과 일관성을 유지하기 위해 데이터 정제 및 가공 작업 수행.  · 크롤링 및 스크래핑 과정에서 발생하는 에러와 예외 처리에 대한 방안 탐구. |
| 10주차  5/1~ | · UI 개발(80%) - HTML,CSS,JS  · Kafka 개발 환경 구축 - Zookeeper, Kafka Server  · 주문 마이크로 서비스 구현(70%) | · 추천 알고리즘에서 사용할 요구 데이터를 수집하기 위한 페이지 화면 구상.  · 페이지의 디자인과 레이아웃을 고려하여 UI/UX 구상(HTML, CSS, JS 기반) |
| 11주차  5/8~ | · Kafka Connect 구축  · Mysql JDBC Connect 연결  · 주문 마이크로 서비스 구현(80%)  · 결제 마이크로 서비스 구현(10%) | · 수집 가능한 데이터의 특성을 파악하고 EDA 기법으로 분석하는 작업 진행. |
| 12주차  5/15~ | · React 주문 기능 연동  · 결제 마이크로 서비스 구현(25%)  · I’mport API 연동 | · 데이터의 특성과 목적을 고려하여 최적의 추천 알고리즘을 선택하기 위해 각 알고리즘의 적합성 검토 |
| 13주차  5/22~ | · 상품 마이크로 서비스 연동 (50%)  · 주문 마이크로 서비스 연동 (90%) | · selenium, BeautifulSoup를 활용한 다나와 웹스크래핑 자동화 구현 및 노트북 데이터셋(이름, 이미지url, 각종 스펙 등) 확보 |
| 14주차  5/29~ | · 장바구니, 상품, 마이페이지 React 변환(90%)  · 상품 마이크로 서비스 연동 (100%)  · 장바구니 마이크로 서비스 연동 (80%) | · Cold-Start문제를 해결하기 위한 초기 AI 추천 알고리즘으로 각각의 스펙 데이터를 모두 합쳐 하나의 문장으로 취급하여 자연어처리(NLP)기법으로 유사도를 측정한 후 추천하는 알고리즘 개발. |
| 15주차  6/5~ | · 장바구니 마이크로 서비스 연동 (100%)  · React 연동 (50%)  · Kafka 연동 (80%) | · Cosine-Similarity 기반 NLP 추천 알고리즘 채택 및 유사도 값을 출력하여 상위 값에 대한 노트북 추천 페이지 구현 |

**5. 개발 결과 (추후에 제출)**

**6. 결론**

**6.1 기대 효과**

**1. 개인 맞춤형 추천**

* 사용자의 선호도, 용도, 예산 등의 다양한 요소를 고려하여 개인 맞춤형 추천을 제공
* 이를 통해 사용자들은 보다 더 적합한 제품을 찾을 수 있으며, 보다 빠르고 간편하게 제품을 선택할 수 있게 되고, 쇼핑에 대한 만족도가 높아지게 됨

**2. 상품 정보의 정확성**

* AI 기술을 활용하여 상품 정보를 자동으로 수집하고 분석
* 이를 통해, 상품 정보의 정확성이 향상되어 사용자들은 보다 더 정확한 정보를 기반으로 제품 선택 가능

**3. 예측 분석**

* 사용자들의 검색 이력 및 구매 이력 등을 분석하여 제품 추천뿐만 아니라, 미래의 쇼핑 트렌드와 사용자들의 선호도 등을 예측 가능
* 이는 더 나은 마케팅 전략을 수립하는 데 도움

**4. 경쟁력 강화**

* AI 기술을 활용한 노트북 추천 사이트는 다른 쇼핑몰과의 경쟁에서 경쟁력을 강화.
* AI 기술을 활용한 맞춤형 추천 서비스는 다른 쇼핑몰과 차별화된 서비스를 제공할 수 있으며, 이는 사용자의 이탈을 막고 유지 보수에도 도움

**6.2 활용 방안**

**1. 고객 맞춤형 추천 시스템**

* 사용자의 선호도, 이전 구매 이력, 검색 패턴 등을 분석하여 개인화된 노트북을 추천함으로써, 고객에게 가장 적합한 제품을 제안할 수 있습니다.

**2. 인공지능 기반 마케팅**

* 사용자들의 구매 이력 및 검색 이력 등을 바탕으로 미래의 쇼핑 트렌드나 사용자 선호도를 예측하여 마케팅 전략을 수립하거나 수정할 수 있습니다. 이를 통해 보다 효과적인 타겟 마케팅이 가능하게 됩니다**.**

**3. 서비스 개선 및 신규 서비스 개발**

* '트북이'를 통해 수집된 사용자 데이터를 분석하여, 서비스 개선점을 파악하고, 이를 통해 서비스 품질을 향상시킬 수 있습니다. 또한, 신규 서비스 아이디어를 도출하거나 사용자의 니즈에 맞는 신규 서비스를 개발하는데 활용될 수 있습니다.

**4. 차별화된 고객 경험 제공**

* 개인화된 추천을 통해 고객에게 차별화된 쇼핑 경험을 제공하고, 이를 통해 고객 만족도를 향상시키며 고객 이탈을 방지할 수 있습니다.

**5. 재구매 유도**

* 고객의 구매 이력과 선호도를 분석하여 개인화된 추천을 제공함으로써, 고객의 재구매를 유도할 수 있습니다. 이는 장기적으로 고객의 일상생활의 가치를 높이는 데에 기여할 것입니다.

**부록**

**[참고 자료 & 문헌 & 도움을 준 사람들]**

[MSA 강의 영상]

<https://www.inflearn.com/roadmaps/409>

Spring Boot & Spring Cloud 마이크로 서비스 애플리케이션 개발(MSA)

[Spring 강의 영상]

<https://www.inflearn.com/roadmaps/149>

스프링부트와 JPA 실무 완전 정복