

홍대 카페 맞춤 추천 서비스



1. 프로젝트 개요

목적:

- 소비자의 취향을 분석하여 최적의 카페를 추천하는 웹서비스를 개발
- 카페 이용 데이터를 바탕으로 맞춤형 추천 서비스를 제공
- 홍대 지역 카페와 소비자 간의 연결을 강화하여 카페 방문율 증대
- 소비자 개별 취향에 맞춘 카페를 추천하는 빅데이터 기반 웹서비스를 제안
- 소비자들의 편의성을 높이고 카페의 고객 유입 증대

1. 프로젝트 개요

기대 효과:

- 소비자 맞춤형 경험 제공
- 카페 방문율 및 고객 유입 증대
- 데이터 기반 의사결정 지원
- 편의성 및 사용자 경험 향상

2. 주요 기능

사용자 맞춤 추천:

- 사용자의 선호도를 바탕으로 한 추천
선호유형: 프랜차이즈 여부, 분위기 등

다른 방식의 추천:

- 리뷰 수, 좋아요 등을 활용한 추천
- 성별, 연령대 별 추천

2. 주요 기능

다양한 방식의 검색기능:

- 키워드 기반의 검색
- 검색된 결과에 필터를 적용

필터 예시:

거리, 연령대,

편의기능(와이파이, 주차, 흡연실, 휠체어 접근성 등),

분위기(가성비좋은, 고급스러운, 예쁜, 격식있는, 이색적인 카페)

3. 기술 스택

Front-End

Markup:

- HTML5

Styling:

- CSS, Bootstrap (4.6.2)

Scripting:

- JavaScript (ES6)

Back-End Infrastructure

Application:

- Spring Framework (5.3.29), MyBatis(3.5.9)

Database:

- MySQL(8.0.21)

Cloud:

- AWS (RDS)

Search & Analysis:

- Python (3.8.8), Elasticsearch(6.5.1), Kibana (6.5.1)

Web Application Server (WAS):

- Apache Tomcat 8.5

Tools

Development:

- Eclipse (STS 3.9.13), Spyder (4.2.5), VS Code, MySQL Workbench 8.0

Version Control:

- GitHub

4. AWS - 목적

1. 프로젝트의 신속하고 효율적인 배포

- AWS의 다양한 서비스와 인프라를 활용해 애플리케이션을 손쉽게 배포하고 관리

2. 강력하고 편리한 보안 관리

- VPC, 보안 그룹 등의 보안 관리 도구로 보안을 손쉽게 구성 가능

3. 동일한 데이터로 개발 및 테스트

- RDS, OpenSearch 등을 사용하여 모든 팀원이 동일한 데이터로 개발 및 테스트 가능

4. AWS - 활용 계획

1. EC2: (Elastic Compute Cloud)

역할: Java와 Python 애플리케이션을 배포하고 실행하는 서버 인스턴스

2. RDS: (Relational Database Service) - MySQL

역할: MySQL 데이터베이스 관리

3. OpenSearch

역할: Elasticsearch 배포 및 관리

4. AWS - 활용 계획

4. S3 (Simple Storage Service)

역할: 파일 저장소

5. VPC (Virtual Private Cloud)

역할: 네트워크 보안, 주소 호스팅 등

4. AWS - 특이사항

1. RDS 우선 적용

- 상대적으로 설정이 간단하고, 동일한 데이터를 제공해주는 RDS를 우선 적용
- 다른 AWS 서비스에 대한 지식이 부족하여 배포하는데 어려움 존재

2. 비용 및 시간 리스크 관리

- 다른 AWS 서비스들을 초기에 도입할 경우 발생할 수 있는 비용과 시간에 대한 리스크를 최소화하기 위해 프리티어로 가능한 RDS만 우선 적용

4. AWS - 특이사항

3. 복합적인 시스템 적용에 따른 난이도 증가

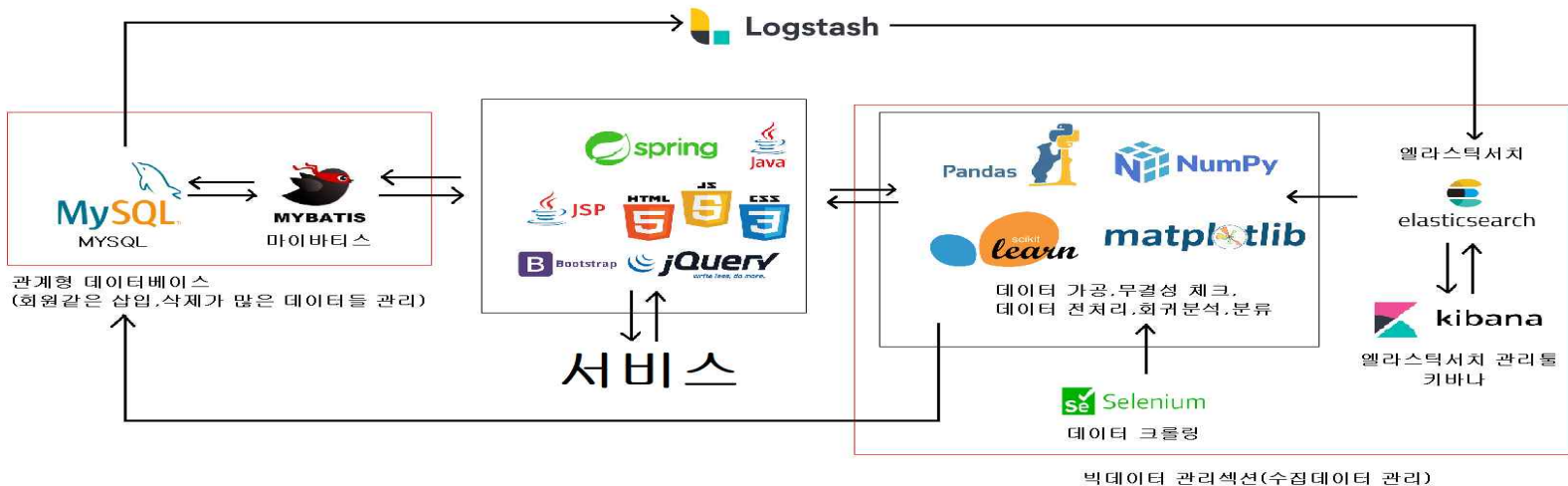
- Java와 DB연결 외에도 Elasticsearch, Python, Logstash 등 다양한 기술이 복합적으로 사용됨에 따라 AWS적용에 더 높은 난이도와 비용이 발생할 가능성 존재

4. 단계적 접근 전략

- 초기 개발 단계에서는 RDS만 적용하고, 개발이 일정 수준 완료된 후 추가적인 AWS 서비스를 도입하여 배포를 시도할 예정

5. 프로젝트 워크플로우

프로젝트 워크플로우



6. 팀 구성 및 역할 분담

PM (Project Manager): 이상현

- 전체 프로젝트 관리 및 일정 조율, 주요 의사결정

PA (Project Assistant): 차지민

- 기술적 리더십 제공, 아키텍처 설계, 코드 품질 관리, 기술적 문제 및 일정상 문제 해결 지원

6. 팀 구성 및 역할 분담 - FRONTEND

PL (Project Leader): 이윤형

- 파트 구성원: 이주연, 백찬혁
- 화면 기획, 설계, 구현
- 화면 정의서
- UI/UX 설계
- 프로토타입
- HTML, CSS, JAVASCRIPT, BOOTSTRAP 사용하여 화면 구현

6. 팀 구성 및 역할 분담 - BACKEND

PL (Project Leader): 차지민

- 파트 구성원: 김은규
- 백엔드 개발: JAVA SPRING FRAMEWORK 기반 API 구현
- 시스템 연동, 구축: 프로젝트 전반적으로 필요한 시스템 연동, 구축, 개선

6. 팀 구성 및 역할 분담 - DATA SCIENCE

PL (Project Leader): 이상현

- 파트 구성원: 이경희
- Selenium, BeautifulSoup: 함께 활용하여 HTML 파싱 및 데이터 추출
- Konlpy (Korean Natural Language Toolkit): 한국어 리뷰 및 텍스트 데이터를 처리하고, 형태소 분석을 통해 키워드 추출, TF-IDF 벡터화와 Logistic Regression을 이용하여 사용자 리뷰의 감성 예측
- Folium, Haversine: 지도 시각화 및 거리 계산을 통해 사용자 위치와 카페 간의 거리 기반 추천 시스템 구현
- Elasticsearch, Kibana: 검색 엔진 구축 및 빠른 데이터 추출을 위한 쿼리 작성
- Database 관리: MySQL, Elasticsearch 데이터 관리

7. 일정/범위관리

Date	Task
8/20 ~ 8/21	주제 선정, 제안서 작성
8/22 ~ 8/26	기능 설계 (Workflow, 기능분해도, CRUD 매트릭스, 메뉴 구성도 등)
8/23 ~ 8/27	시스템 설계 및 테스트
8/23 ~ 8/27	UI / UX 기획
8/28 ~ 8/29	Database 설계
8/28 ~ 8/30	데이터 수집 / 정의 / 가공
9/2 ~ 9/13	개발
9/12 ~ 9/13	통합 및 테스트
9/20 ~ 9/23	보고서 작성

[illegible]

9. 프로젝트 수행 경과

- UI 기획

목적:

1. 기능 명세:

- 실제 UI 화면에서 각 기능이 어떻게 배치되고 동작하는지에 대한 명세
- 사용자 인터랙션 흐름과 화면 간 전환, 버튼 클릭 시 발생하는 이벤트 등 구체적 정의

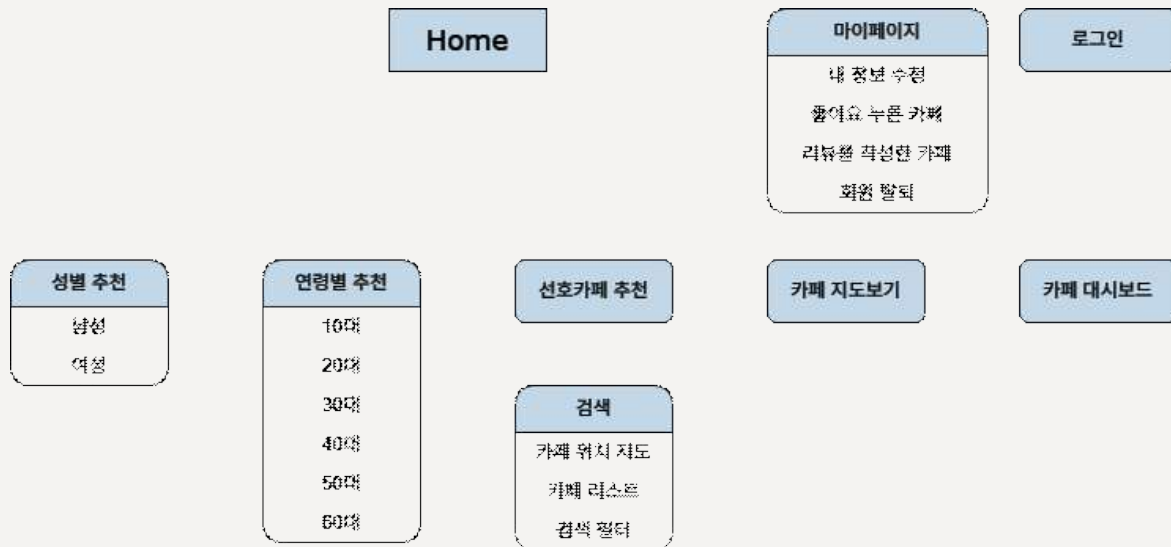
2. 디자인 가이드라인:

- 사용자 인터페이스(UI)의 레이아웃 및 시각적 요소에 대한 가이드라인

9. 프로젝트 수행 경과

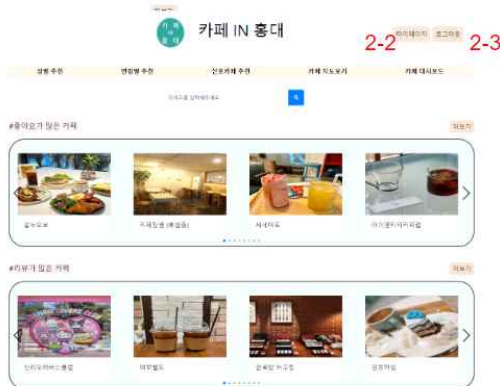
- UI 기획 (메뉴 구성도)

메뉴 구조 (2Depths)



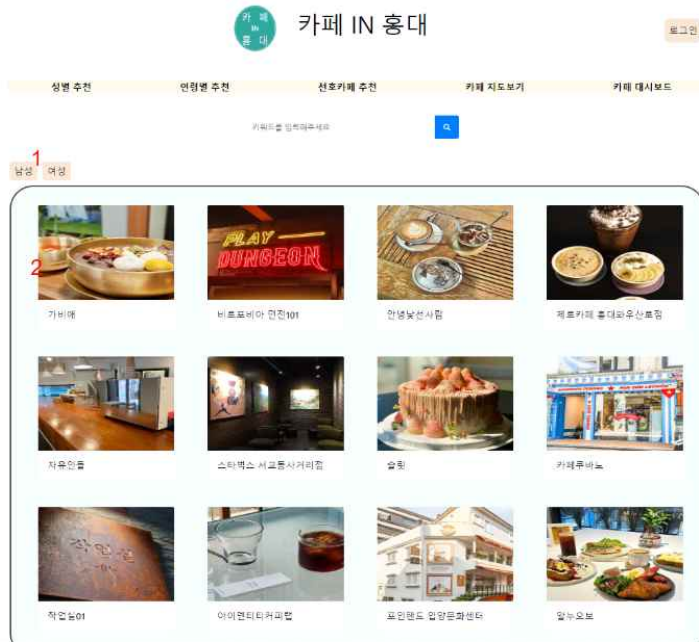
9. 프로젝트 수행 경과

- 기획



1. 로고 클릭시 메인페이지로 이동
- 2-1. 로그인 버튼 클릭시 로그인 페이지로 이동
- 2-2. 로그인시 화면이 바뀌고 마이페이지 클릭시 마이페이지로 이동
- 2-3. 로그아웃 버튼 클릭시 로그아웃되며 메인 페이지로 이동
3. 성별 추천 버튼을 누르면 성별 추천 페이지로 이동
4. 연영별 추천 버튼을 누르면 연영별 추천 페이지로 이동
5. 선호카페 추천 버튼을 누르면 선호카페 추천 페이지로 이동(로그인 필요)
6. 카페 지도 보기 버튼을 누르면 카페 지도가 나옴
7. 카페 대시보드 버튼을 누르면 카페 대시보드가 나옴
8. 더보기 버튼을 누르면 카페 목록이 더 표기됨
9. "<," ">" 표시를 누르면 왼쪽, 오른쪽으로 이동하며 카페 목록이 나옴
10. 카페 사진을 누르면 카페 상세페이지로 이동

9. 프로젝트 수행 경과 - 니 기획

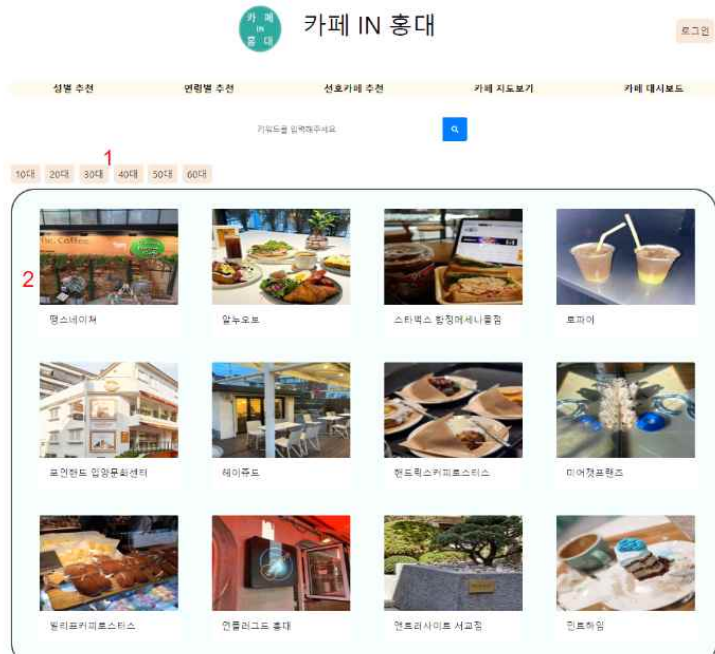


1. 남성,여성 버튼을 누르면
해당 성별 추천

2. 카페 이미지를 클릭시
해당 카페 상세페이지로
이동

9. 프로젝트 수행 경과

- 기획

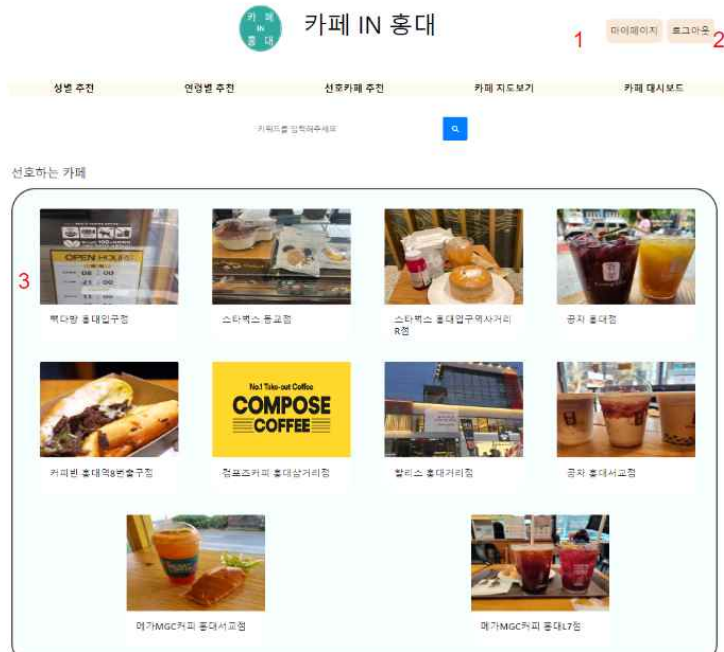


1. 나이대를 클릭하면 해당 나이대별 추천 카페 목록이 나옴

2. 카페 이미지를 클릭하면 카페 상세페이지로 이동

9. 프로젝트 수행 경과

- 기획



1. 마이페이지 버튼을
클릭시 마이페이지
페이지로 이동

2. 로그아웃 버튼을 클릭시
로그아웃 되며
메인페이지로 이동

3. 카페 이미지를 클릭하면
카페 상세페이지로 이동

9. 프로젝트 수행 경과 - 니 기획


카페 IN 홍대

로그인

성별 주전 연령별 주전 선호카페 주전 카페 지도보기 카페 내시보

4 카테고리를 입력해주세요

거리

100 미터 카페

200 미터 카페

300 미터 카페

400 미터 카페

500 미터 카페

600 미터 카페

700 미터 카페

800 미터 카페

900 미터 카페

1000 미터 카페

1100 미터 카페

1200 미터 카페

연령대

10대

20대

30대

40대

50대

60대

편의기능

외아파트

불안실

활제어

주차

반려동물

불이탈

분위기


가성비좋음

가끔스러운

예쁜

작적있음

이색적인



2

3




카페 IN 홍대점

리뷰점수 83.3점



카페 지음아기

리뷰점수 64.3점



홍대반자카페 분점

리뷰점수 73.7점



카페점점 (홍대점)

리뷰점수 95.8점



카페 333

리뷰점수 88.9점



카페 데코믹스 홍대점

리뷰점수 88.0점

1. 거리, 연령별, 편의기능, 분위기에 해당하는 목록들 클릭시 해당하는 카페 목록이 나옴

2. 좋아요 많은순, 리뷰 많은순 등 해당 버튼을 클릭하면 해당하는 카페 목록이 나옴

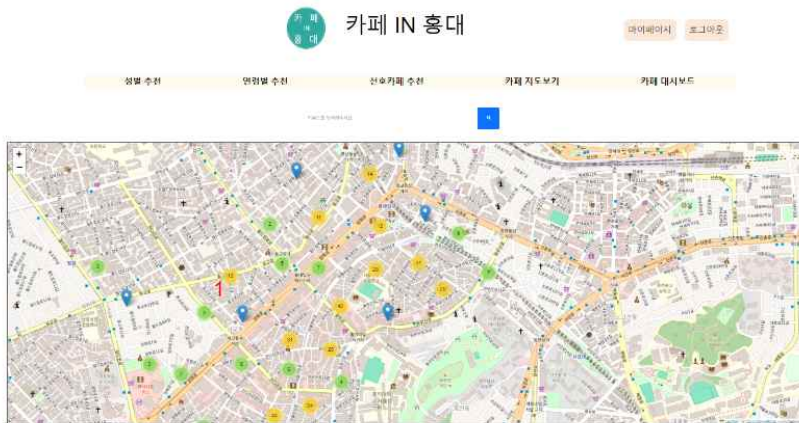
3. 카페 이미지를 클릭하면 카페 상세페이지로 이동

4. 검색어 입력시 해당하는 검색어가 없으면 검색 결과가 없습니다. 문구가 나오고, 해당하는 검색어가 있으면 해당 카페 목록이 나오고 검색어를 입력하지 않으면 모든 카페리스트가 나옴

9. 프로젝트 수행 경과

- 기획

1. 마커 클릭시 해당하는
카페 상세페이지로 이동



9. 프로젝트 수행 경과

- 니 기획



카페 IN 홍대

로그인

성별 추천

연령별 추천

선호카페 추천

카페 지도보기

카페 대시보드

카카오 로그인

로그인

이메일

1 비밀번호

2

3

4

5

1. 이메일과 비밀번호를 기재할 수 있음
2. 로그인 버튼 클릭시 일치하면 로그인이 되고 일치하지 않으면 아이디 또는 비밀번호가 틀리다는 문구가 나옴
3. 회원가입 버튼을 누르면 회원가입 페이지로 이동
4. 아이디 찾기 버튼을 누르면 아이디 찾기 페이지로 이동
5. 비밀번호 찾기 버튼을 누르면 비밀번호 찾기 페이지로 이동

9. 프로젝트 수행 경과

- 기획

카페
IN
홍대

로그인

성별 추천

연령별 추천

선호카페 추천

카페 지도보기

카페 대시보드

카테고리 선택해주세요

회원가입

이메일

1

비밀번호

비밀번호

성명

성별

2

선호하는 카페 유형을 골라주세요

☐ 휴먼사이드 ☐ 가인

3

분위기

☐ 자연적 숲속 카페

☐ 고급스러운 카페

☐ 예쁜 카페

4

☐ 독서있는 카페

☐ 리스펙한 카페

5

가입하기

1. 이메일 등 기재할 수 있음

2. 성별별로 클릭하는 기능


3. 선호하는 카페별로 클릭하는 기능

4. 선호하는 분위기별로 클릭하는 기능

5. 가입하기 버튼을 클릭하면 메인 페이지로 이동

9. 프로젝트 수행 경과

- 기획



카페 IN 홍대

로그인

성별 추천

연령별 추천

선호카페 추천

카페 지도보기

카페 대시보드

카테고리를 입력해주세요

아이디 찾기

이름

휴대폰번호

아이디 찾기

비밀번호 찾기

로그인

1. 이름과 휴대폰 번호를
기재할 수 있음

2. 아이디 찾기 버튼을
누르면 새로고침 됨

3. 비밀번호 찾기 버튼을
누르면 비밀번호 찾기
페이지로 이동

4. 로그인 버튼을 누르면
로그인 페이지로 이동

9. 프로젝트 수행 경과

- 기획



카페 IN 홍대

로그인

성별 추천

연령별 추천

선호카페 추천

카페 지도보기

카페 대시보드

카카오 로그인

비밀번호 찾기

이메일

이름

비밀번호 찾기

아이디 찾기

로그인

1. 이메일과 이름을 기재할 수 있음
2. 비밀번호 찾기 버튼을 누르면 새로고침 됨
3. 아이디 찾기 버튼을 누르면 아이디 찾기 페이지로 이동
4. 로그인 버튼을 누르면 로그인 페이지로 이동

9. 프로젝트 수행 경과

- 기획

카페 IN 홍대

로그아웃

성별 추천 연립별 추천 선호카페 추천 카페 지도보기 카페 내시보

카드로 결제하기

1. 내 정보 수정

나의 활동

- 활동한 카페
- 활동한 카페
- 활동한 카페

2. 내 정보 수정

이메일: user1@naver.com

비밀번호: ****

이름: 김민준

전화번호: 010-7234-5678

생년월일: 1999-12-12

성별: ☒ 남 ☐ 여

3. 선호하는 카페

4. 선호하는 분위기

5. 선호하는 분위기

6. 수정

1. 해당 글자를 누르면 해당되는 내 정보 수정, 좋아요 누른 카페, 리뷰 작성한 카페, 회원탈퇴 페이지로 이동함

2. 내정보를 수정하여 기재할 수 있음

3. 성별별로 클릭하는 기능

4. 선호하는 카페별로 클릭하는 기능

5. 선호하는 분위기별로 클릭하는 기능

6. 수정하기 버튼을 클릭하면 수정된 것을 반영하여 새로고침 됨

9. 프로젝트 수행 경과

- 기획

가
매
홍
대

카페 IN 홍대

마이페이지

로그아웃

성별 추천

연령별 추천

선호카페 추천

카페 지도보기

카페 대시보드

카테고리를 입력해주세요

내 정보 수정

나의 활동

종이로 누른 카피

리뷰를 작성한 카페

회원탈퇴

회원탈퇴

비밀번호 입력 시 회원탈퇴가 완료됩니다. (나)

1 비밀번호를 입력하세요

2

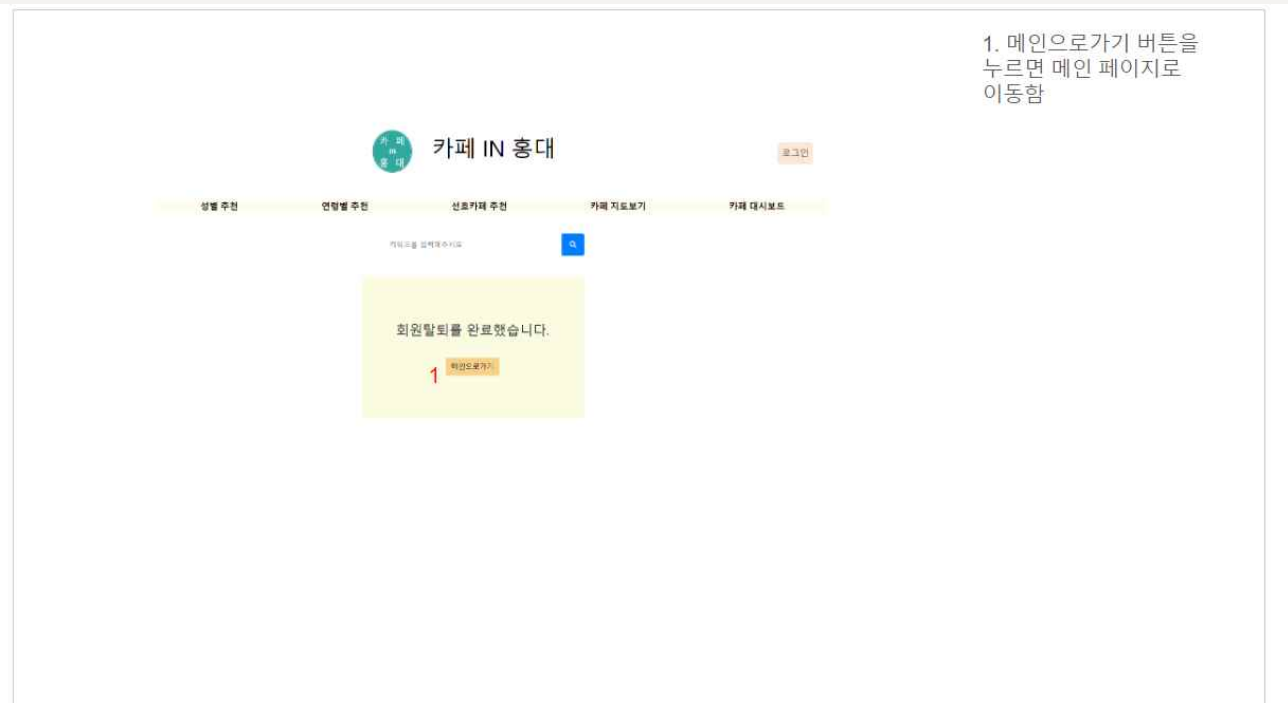
탈퇴

1. 탈퇴할 계정의
비밀번호를 기재할 수 있음

2. 탈퇴 버튼을 누르면
탈퇴가 완료되었다는
문구가 나오는 페이지가
나옴

9. 프로젝트 수행 경과

- UI 기획



9. 프로젝트 수행 경과

- 기획



1. 사진을 클릭하면 해당 카페의 상세페이지로 이동

2. 하트 버튼을 누르면 좋아요가 취소됨

3. 버튼을 누르면 해당하는 페이지로 이동

9. 프로젝트 수행 경과

- 기획

카
페
IN
홍
대

카페 IN 홍대

마이페이지로그아웃

성별 추천

연령별 추천

선호카페 추천

카페 지도보기


카페 대시보드

카테고리: 일례적유자료

내 정보 수정:
나의 활동
• 좋아한 카페
• 리뷰를 작성한 카페
• 활동활동

리뷰를 작성한 카페


1



영문도 5.0점
21534543

수정삭제

2



비프코비어 단한1014.0점
235234

수정삭제

3

1

2

3

4

1. 사진을 클릭하면 해당 카페의 상세페이지로 이동

2. 수정 버튼을 누르면 리뷰를 수정하는 페이지로 이동

3. 삭제 버튼을 누르면 작성한 리뷰가 삭제됨

4. 버튼을 누르면 해당하는 페이지로 이동

9. 프로젝트 수행 경과 - 기획



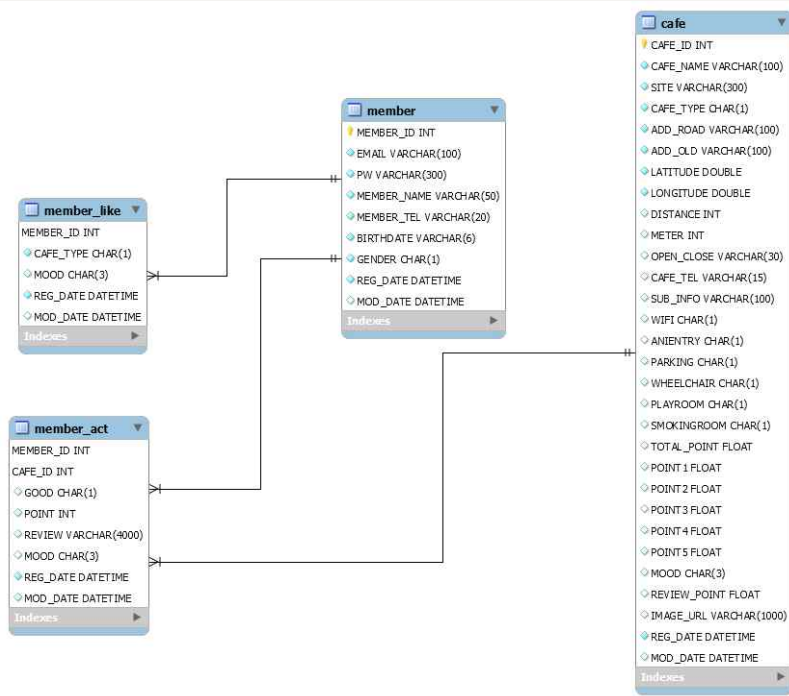
1. 하트 버튼을 누르면
좋아요가 되고 좋아요가
되어 있다면 취소됨

2. 리뷰 작성하기 버튼을
누르면 리뷰 작성하는
페이지로 이동

3. 스크롤 바를 내리거나
올리면 작성되어있는
리뷰가 보임

9. 프로젝트 수행 경과

- ERO



9. 프로젝트 수행 경과

- 테이블 정의서

테이블 정의서											
시스템명		홍대 카페 맞춤 추천 웹서비스								서브시스템명	
작성일		2024.09.02								작성자	
DATABASE ID										테이블 ID	MEMBER
테이블명		회원								PAGE	1/4
개요		회원 정보 관리									
일련 번호	한글 컬럼명	영문 컬럼명	DATA TYPE	LENGTH	PK	FK	INDEX	NULL	UNIQUE	DEFAULT	DEFINITION
1	사용자고유번호	MEMBER_ID	INT		Y		Y	N	Y	AUTO_INCREMENT	사용자 고유 번호 (숫자형)
2	이메일	EMAIL	VARCHAR	100				N	Y		
3	비밀번호	PW	VARCHAR	300				N			
4	성명	MEMBER_NAME	VARCHAR	50				N			
5	전화번호	MEMBER_TEL	VARCHAR	20				N			
6	생년월일	BIRTHDATE	VARCHAR	6				N			사용자 생년월일 (YYMMDD)
7	성별	GENDER	CHAR	1				N			사용자 성별 ('M', 'F')
8	가입일자	REG_DATE	DATETIME					N		NOW()	가입 날짜 및 시간
9	수정일자	MOD_DATE	DATETIME					Y		NOW()	마지막 수정 날짜 및 시간

9. 프로젝트 수행 경과

- 테이블 정의서

테이블 정의서											
시 스템 명		홍대 카페 맞춤 추천 웹서비스								서비스시스템명	
작 성 일		2024.09.02								작성자	
DATABASE ID										테이블 ID	CAFE
테이블명		카페								PAGE	2/4
개요		카페 정보 관리									
일련 번호	한글 컬럼명	영문 컬럼명	DATA TYPE	LEN GTH	PK	FK	INDEX	NUL L	UNI QUE	DEFAULT	DEFINITION
1	카페고유번호	CAFE_ID	INT		Y		Y	N	Y	AUTO_INCREMENT	카페 고유 번호 (숫자형)
2	카페이름	CAFE_NAME	VARCHAR	100				N			
3	SITE	SITE	VARCHAR	300				N			
4	카페유형	CAFE_TYPE	CHAR	1				N			F : 프랜차이즈, P : 개인
5	주소	ADD_ROAD	VARCHAR	100				N			
6	소재지번	ADD_OLD	VARCHAR	100				N			
7	위도	LATITUDE	DOUBLE					N			
8	경도	LONGITUDE	DOUBLE					N			
9	거리	DISTANCE	INT					Y			홍대입구역에서 거리
10	거리단위	METER	INT					Y			홍대입구역에서 미터단위
11	영업시간	OPEN_CLOSE	VARCHAR	30				Y			
12	전화번호	CAFE_TEL	VARCHAR	15				Y			
13	부가정보	SUB_INFO	VARCHAR	100				Y			
14	와이파이	WIFI	CHAR	1				Y			Y : 가능, N : 불가

9. 프로젝트 수행 경과

- 테이블 정의서

15	동물출입	ANIENTRY	CHAR	1				Y			Y : 가능 , N : 불가
16	주차	PARKING	CHAR	1				Y			Y : 가능 , N : 불가
17	휠체어사용	WHEELCHAIR	CHAR	1				Y			Y : 가능 , N : 불가
18	놀이방	PLAYROOM	CHAR	1				Y			Y : 가능 , N : 불가
19	흡연실	SMOKINGROOM	CHAR	1				Y			Y : 가능 , N : 불가
20	총점수	TOTAL_POINT	INT					Y			
21	맛점수	POINT1	INT					Y			
22	친절점수	POINT2	INT					Y			
23	분위기점수	POINT3	INT					Y			
24	가성비점수	POINT4	INT					Y			
25	주차점수	POINT5	INT					Y			
26	분위기	MOOD	CHAR	3				Y			M02: 고급스러운 M03: 예쁜 M04: 객실있는 M05: 이색적인 M01: 가성비좋은
27	리뷰점수	REVIEW_POINT	FLOAT					Y			해당 카페 리뷰 감성 분석 점수
28	카페이미지	IMAGE_URL	VARCHAR	1000				Y			
29	등록일자	REG_DATE	DATETIME					N			
30	수정일자	MOD_DATE	DATETIME					Y			

9. 프로젝트 수행 경과

- 테이블 정의서

테이블 정의서											
시스템명		홍대 카페 맞춤 추천 웹서비스								서브시스템명	
작성일		2024.09.02								작성자	
DATABASE ID										테이블 ID	MEMBER_ACT
테이블명		회원활동								PAGE	3/4
개요		회원 활동 관리									
일련 번호	한글 컬럼명	영문 컬럼명	DATA TYPE	LENGT H	PK	FK	INDEX	NULL	UNIQUE	DEFAULT	DEFINITION
1	카페고유번호	CAFE_ID	INT		Y	Y	Y	N			
2	사용자고유번호	MEMBER_ID	INT		Y	Y	Y	N			
3	좋아요	GOOD	CHAR(1)					Y			Y: 좋아요
4	별점	POINT	INT					Y		NULL	
5	리뷰	REVIEW	VARCHAR	4000				Y		NULL	
6	분위기	MOOD	CHAR	3				Y			M02: 고급스러운 M03: 예쁜 M04: 격식있는 M05: 이색적인 M01: 가성비좋은
7	등록일자	REG_DATE	DATETIME					N		NOW()	
8	리뷰수정일자	MOD_DATE	DATETIME					Y		NOW()	

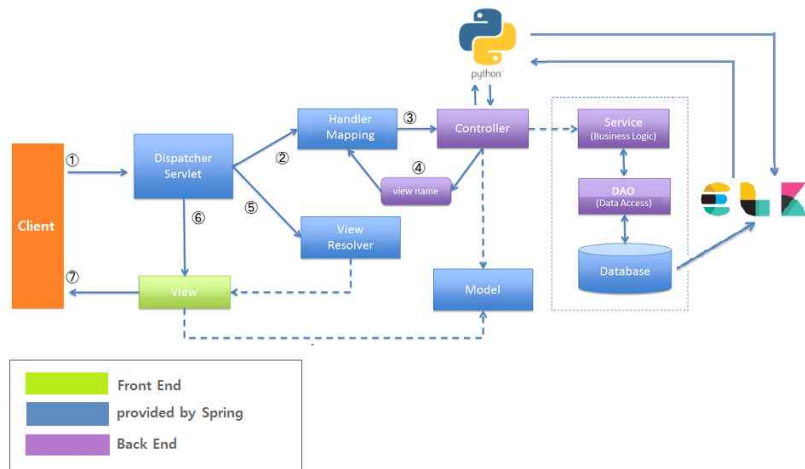
9. 프로젝트 수행 경과

- 테이블 정의서

테이블 정의서											
시 스템 명		홍대 카페 맞춤 추천 웹서비스								서브시스템명	
작 성 일		2024.09.02								작성자	
DATABASE ID										테이블 ID	MEMBER_LIKE
테이블명		선호유형								PAGE	4/4
개요											
일련 번호	한글 컬럼명	영문 컬럼명	DATA TYPE	LENGTH	PK	FK	INDEX	NULL	UNIQUE	DEFAULT	DEFINITION
1	사용자고유번호	MEMBER_ID	INT		Y	Y	Y	N			사용자 고유 번호 (사용자 정보 FK)
2	카페유형	CAFE_TYPE	CHAR	1				N			선호하는 카페 유형 (F: 프랜차이즈, P: 개인)
3	분위기	MOOD	CHAR	3				Y			M01: 가성비좋은 M02: 고급스러운 M03: 예쁜 M04: 격식있는 M05: 이색적인
4	등록일자	REG_DATE	DATETIME					N		NOW()	선호 정보 등록 날짜 및 시간
5	수정일자	MOD_DATE	DATETIME					Y		NOW()	선호 정보 수정 날짜 및 시간

9. 프로젝트 수행 경과

- Software Architecture



9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

추천 및 검색 프로세스 (성별 추천/연령대별 추천/검색)

Step 1 (JAVA): 추천에서 남성/여성/연령대 선택 및 검색시 검색어 입력

Step 2 (JAVA -> Python): Step1의 정보를 Python으로 전달

Step 3 (Python): 전달받은 정보를 바탕으로 Elasticsearch 쿼리를 생성

Step 4 (ElasticSearch): 생성된 쿼리를 Elasticsearch에 전송하여 검색을 실행

Step 5 (Python): 검색 결과를 받아 필요한 필드를 추출

Step 6 (Python -> JAVA): 추출한 결과를 Java로 전송

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링

The logo for BeautifulSoup, featuring the word "Beautiful" in a standard sans-serif font and "Soup" in a stylized script font, with a small integral symbol (∫) replacing the letter 'o'.

웹 페이지의 HTML 코드를 파싱하여 필요한 정보(텍스트, 태그, 속성 등)를 추출하는 데 사용되는 라이브러리. 다만, 자바스크립트로 동적 생성된 콘텐츠는 처리하지 못함

The Selenium logo, consisting of a green square with a white checkmark and the letters "se" in white, followed by the word "Selenium" in a green sans-serif font.

브라우저 자동화 도구로, 자바스크립트로 동적으로 생성되는 웹 콘텐츠도 처리할 수 있음. 페이지를 탐색하고, 버튼 클릭, 스크롤, 로그인 같은 사용자 행동을 자동화하여 동적인 웹 페이지에서 데이터를 수집함

카페 데이터를 수집하려 했던 카카오맵은 자바스크립트를 사용하는 동적 로딩 방식이 많기 때문에 selenium을 사용하여 동적 콘텐츠를 로드해야만 함. 다만, 동적 콘텐츠를 로드한 후 데이터 수집은 BeautifulSoup이 더 효율적이기 때문에 복합 사용함.

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링 - 카페 리스트

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.options import Options
import pandas as pd
import time

chrome_options = Options()
driver = webdriver.Chrome(options = chrome_options)

# 카카오맵 웹사이트 열기
driver.get('https://map.kakao.com/')

#search\keyword\query

# 검색창에 '홍대카페' 입력하고 검색
search_box = driver.find_element(By.CSS_SELECTOR, '#search\keyword\query')
search_box.send_keys('홍대카페')
search_box.send_keys(Keys.RETURN)

# 페이지 로딩 대기
time.sleep(5)

results = []
```

```
def get_content(driver):

    # 페이지 소스 가져오기
    page_source = driver.page_source

    # BeautifulSoup으로 HTML 파싱
    soup = BeautifulSoup(page_source, 'html.parser')

    # '카페 이름 a.link_name'과 '상세 a.moreview' 태그 추출
    cafe_names = soup.select('a.link_name')
    more_view = soup.select('a.moreview')

    # 결과 출력
    for name, view in zip(cafe_names, more_views):
        print(f"카페 이름: {name.text.strip()}")
        print(f"상세보 링크: {view['href']}")
        print("-"*50)

        results.append([name.text.strip(), view['href']])

def move(driver):
    right = driver.find_elements('css selector', '#info\.search\.place\.more')[0]
    right.click()
    time.sleep(5)

def move_next(driver, cnt):
    print(">>>>>>>>>>>>>>>> cnt :", cnt)
    tag = '#info\.search\.page\.no' + str(cnt)
    right = driver.find_elements('css selector', tag)[0]
    right.click()
    time.sleep(5)

def move_nextFlag(driver):
    right = driver.find_elements('css selector', '#info\.search\.page\.next')[0]
    right.click()
    time.sleep(5)
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링 - 카페 상세정보

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.options import Options
import pandas as pd

df = pd.read_csv('cafeInfo.csv')

chrome_options = Options()
driver = webdriver.Chrome(options = chrome_options)

res = []
```

```
def get_content(driver, name, link) :

    # 주소
    addFind = driver.find_elements(By.CSS_SELECTOR, '#mArticle > div.cont_essential > div.details_placeinfo > div:nth-child(2) > div > span.txt_address')
    #for addV in addFind: pass
    if not addFind: addTran = 'none'
    else:
        for addV in addFind: addTran = addV.text

    # 소재지번
    oldAddFind = driver.find_elements(By.CSS_SELECTOR, '#mArticle > div.cont_essential > div.details_placeinfo > div:nth-child(2) > div > span.txt_addrnum')
    if not oldAddFind: oldAddTran = 'none'
    else:
        for oldAddV in oldAddFind:
            oldAddTran = oldAddV.text[2:]

    # 영업시간
    businessHours = driver.find_elements(By.CSS_SELECTOR, '#mArticle > div.cont_essential > div.details_placeinfo > div:nth-child(3) > div > div > ul > li > span > span')
    if not businessHours: businessHoursTran = 'none'
    else:
        for businessHoursV in businessHours: businessHoursTran = businessHoursV.text

    # 전화번호
    tel = driver.find_elements(By.CSS_SELECTOR, '#mArticle > div.cont_essential > div.details_placeinfo > div.placeinfo_default.placeinfo_contact > div > div > span > span.txt_c')
    if not tel: telTran = 'none'
    else:
        for telV in tel: telTran = telV.text

    # 부가정보 (예약가능, 배달불가, 주차가능)
    info1 = driver.find_elements(By.CSS_SELECTOR, '#mArticle > div.cont_essential > div.details_placeinfo > div:nth-child(7) > div')
    if not info1: info1Tran = 'none'
    else:
        for info1V in info1: info1Tran = info1V.text

    # 편의시설 (wifi, 동물출입, 주차, 휠체어사용, 놀이방, 흡연실)
    # 정보를 가져 올 수 없는 관계로 dummy 데이터를 빈으로 생성한다. -> 열설에서 상세 수정한다.
    info2_wifiTran = '와이파이'
    info2_anEntryTran = '동물출입'
    info2_parkingTran = '주차'
    info2_wheelchairTran = '휠체어사용'
    info2_playroomTran = '놀이방'
    info2_smokingRoomTran = '흡연실'
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링 - 카페 상세정보

```
for index, row in df[["name", "link"]].iterrows():
    name = row["name"]
    link = row["link"]
    print (name, link)
    driver.get(link)
    time.sleep(5)
    get_content(driver, name, link)

sum_df = pd.DataFrame(res)

sum_df.columns = ['카페명', 'SITE', '주소', '소재지번', '영업시간', '전화번호',
                  '부가정보', '와이파이', '동물출입', '주차', '휠체어사용', '놀이방', '흡연실',
                  '총점수', '맛점수', '친절점수', '분위기점수', '가성비점수', '주차점수', '카페이미지']

sum_df.to_csv('cafeInfoDetail.csv')
```


9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링 - 카페 리뷰

```
from opensearchpy import OpenSearch
import csv
import json
import sys

# UTF-8 인코딩 설정 (파일에서 실행 시 주석처리 해야함, 자바에서 사용시 주석 해제)
# sys.stdout.reconfigure(encoding='utf-8')

# Elasticsearch 클러스터에 연결
try:
    es = OpenSearch('http://localhost:9200')
except Exception as e:
    print(f"Error connecting to OpenSearch: {e}")
    exit(1)

# Step 1: 모든 cafe_name 가져오기
query = {
    "query": {
        "match_all": {}
    },
    "_source": ["cafe_id", "cafe_name", "site"], # cafe_id, cafe_name과 site만 추출
    "size": 10000 # 필요에 따라 조정, 기본값은 10,000
}

# 검색
response = es.search(index="vw_cafe_index", body=query, scroll='2m')

# 전체 문서 수집
all_cafe_data = []
scroll_id = response['_scroll_id']
```

```
while True:
    # 현재 스크롤 페이지의 문서 수
    hits = response['hits']['hits']
    if not hits:
        break

    # 문서에서 cafe_id, cafe_name과 site 추출
    for hit in hits:
        cafe_id = hit['_source'].get('cafe_id')
        cafe_name = hit['_source'].get('cafe_name')
        site = hit['_source'].get('site')
        if cafe_name:
            all_cafe_data.append([cafe_id, cafe_name, site])

    # 다음 페이지로 이동
    response = es.scroll(scroll_id=scroll_id, scroll='2m')

# 마지막 스크롤 요청 종료
es.clear_scroll(scroll_id=scroll_id)

# Step 2: CSV 파일로 저장
csv_file = 'cafe_data.csv'
with open(csv_file, mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    writer.writerow(['cafe_id', 'cafe_name', 'site']) # 헤더 작성
    writer.writerows(all_cafe_data) # 모든 cafe_id, cafe_name과 site 데이터 작성

print(f"Saved {len(all_cafe_data)} cafe IDs, names, and sites to {csv_file}.")
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

1. 카페 데이터 크롤링 - 카페 리뷰

```
# 각 카페 URL로 접속하여 리뷰 데이터 크롤링
for cafe_id, cafe_name, site in cafes:
    driver.get(site)
    time.sleep(3)

    try:
        # 리뷰 섹션 찾기
        review_section = driver.find_element_by_class_name('evaluation_review')
    except Exception as e:
        print(f"리뷰 섹션을 찾을 수 없습니다. 카페 ID {cafe_id}에 리뷰가 없을 가능성이 있습니다: {e}")
        continue # 다음 카페로 넘어감

    # 리뷰 섹션 내에서 더보기 버튼을 최대 5번 클릭하여 더 많은 리뷰를 로드
    for _ in range(5):
        try:
            # 리뷰 섹션 내에서 더보기 버튼 찾기
            more_reviews_button = review_section.find_element_by_xpath('//a[contains(@class, "link_more")]')

            # "추가 보기" 버튼이 나타나면 반복 중단
            if "link_unfold" in more_reviews_button.get_attribute("class"):
                print("추가 보기 버튼이 나타났습니다. 더 이상 클릭하지 않습니다.")
                break

            # 더보기 버튼 클릭
            more_reviews_button.click()
            time.sleep(2)
        except Exception as e:
            print(f"추가 더보기 버튼을 찾거나 클릭하는 데 실패했습니다: {e}")
            break

    # 리뷰, 별점, 좋았던 점 등 클래스 이름으로 추출
    review_elements = review_section.find_elements_by_class_name('text_comment')
    star_elements = review_section.find_elements_by_class_name('inner_star')
    like_point_elements = review_section.find_elements_by_class_name('group_likepoint')

    # 리뷰가 없으면 다음 카페로 넘어감
    if len(review_elements) == 0:
        print(f"카페 ID {cafe_id}에 리뷰가 없습니다.")
        continue

    # 각 리스트 길이를 확인하여 데이터 처리
    num_reviews = len(review_elements)
    num_stars = len(star_elements)
    num_like_points = len(like_point_elements)
```

```
# 각 리스트 길이를 확인하여 데이터 처리
num_reviews = len(review_elements)
num_stars = len(star_elements)
num_like_points = len(like_point_elements)

for i in range(num_reviews):
    try:
        review_text = review_elements[i].find_element_by_tag_name('span').text

        # 별점의 width 값을 추출, 변환
        star_style = star_elements[i].get_attribute('style')
        star_width = star_style.split(':')[1].strip().replace('%', '')
        star_percentage = float(star_width) / 100 # 피센트를 숫자로 변환
        rating = star_percentage / 20.0 # 피센트를 5점 만점으로 변환

        # 좋았던 점 추출
        if i < num_like_points:
            like_point_texts = [point.text for point in like_point_elements[i].find_elements_by_class_name('chip_likepoint')]
            like_point_code = f"M{random.randint(1, 5):02d}" if like_point_texts else 'null'
        else:
            like_point_code = 'null'

        # 결과 저장
        results.append([cafe_id, rating, review_text, like_point_code])

    # 결과 출력
    print(f"카페 ID: {cafe_id}")
    print(f"리뷰: {review_text}")
    print(f"별점: {rating}점")
    print(f"좋았던 점 코드: {like_point_code}")
    print(f"{'-' * 50}")

except Exception as e:
    print(f"데이터 추출에 실패했습니다: {e}")
    continue

# 결과를 CSV 파일로 저장
with open('cafe_reviews.csv', mode='w', encoding='utf-8', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['cafe_id', 'point', 'review', 'mood'])
    writer.writerows(results)
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

2. 카페 리뷰 데이터 감성 분석



한국어 자연어 처리를 위한 라이브러리로, 형태소 분석, 품사 태깅, 텍스트 전처리 등을 지원



머신러닝을 위한 라이브러리로, 데이터 전처리, 모델 학습 및 평가 등을 지원

`TfidfVectorizer`를 사용하여 텍스트 데이터를 TF-IDF 방식으로 벡터화. 이를 통해 단어의 중요도를 반영한 피처를 생성

`train_test_split()`를 사용하여 데이터를 학습용과 테스트용으로 분리

`LogisticRegression` 클래스를 사용하여 로지스틱 회귀 모델을 학습하고, 이를 통해 감성을 예측

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

2. 카페 리뷰 데이터 감성 분석

#2% naver 영화 리뷰 데이터로 학습 후 카페 리뷰 감성분석

```
import mysql.connector
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from konpy.tag import Okt
```

```
# 1. ratings_train.txt 데이터 불러오기 및 전처리
df_train = pd.read_csv('./data/ratings_train.txt', sep='\t')
```

```
# 결측치 제거
df_train = df_train.dropna(subset=['document'])
```

```
# 엑스트라 정제 (특수문자 및 공백 제거)
df_train['document'] = df_train['document'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "", regex=True)
df_train = df_train[df_train['document'].str.strip().astype(bool)]
```

```
# 형태소 분석
okt = Okt()
df_train['tokenized'] = df_train['document'].apply(okt.morphs)
df_train['tokenized'] = df_train['tokenized'].apply(lambda x: ' '.join(x))
```

```
# TF-IDF 벡터화
tfidf = TfidfVectorizer()
X_train_data = tfidf.fit_transform(df_train['tokenized'])
y_train_data = df_train['label'] # 긍정(1), 부정(0)
```

```
# 데이터 분리
X_train, X_test, y_train, y_test = train_test_split(X_train_data, y_train_data, test_size=0.2, random_state=42)
```

```
# 로지스틱 회귀 모델 학습
model = LogisticRegression()
model.fit(X_train, y_train)
```

2. MySQL에서 카페 리뷰 데이터를 불러오기 및 전처리

```
conn = mysql.connector.connect(
    host='192.168.0.26',
    user='user',
    password='mysql',
    database='main_project'
)
```

```
query = """
SELECT cafe_id, point, review
FROM member_act
WHERE review IS NOT NULL AND review != ''
"""
df_cafe = pd.read_sql(query, conn)
conn.close()
```

```
# 결측치 처리 및 특수문자 제거
df_cafe['review'] = df_cafe['review'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "", regex=True)
df_cafe = df_cafe[df_cafe['review'].str.strip().astype(bool)]
```

```
# 형태소 분석 및 토큰화
df_cafe['tokenized'] = df_cafe['review'].apply(okt.morphs)
df_cafe['tokenized'] = df_cafe['tokenized'].apply(lambda x: ' '.join(x))
```

```
# 카페 리뷰 데이터도 TF-IDF로 벡터화
X_cafe = tfidf.transform(df_cafe['tokenized']) # 동일한 TF-IDF 벡터라이저 사용
```

```
# 학습된 모델을 사용해 카페 리뷰 데이터의 감성 예측
df_cafe['text_sentiment'] = model.predict(X_cafe) # 1은 긍정, 0은 부정
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

2. 카페 리뷰 데이터 감성 분석

```
# 3. 감성 분석 결과와 점수를 결합한 레이블 생성
def combined_sentiment(text_sentiment, point):
    if point >= 4.0 and text_sentiment == 1:
        return 1 # 긍정
    elif point <= 2.0 and text_sentiment == 0:
        return 0 # 부정
    else:
        if point >= 4.0:
            return 1 # 높은 점수로 긍정 판단
        elif point <= 2.0:
            return 0 # 낮은 점수로 부정 판단
        else:
            return text_sentiment # 중간 점수는 텍스트 분석 결과에 따름

# 카페 아이디별 감성 분석 적용
def apply_combined_sentiment(group):
    group['combined_sentiment'] = group.apply(lambda x: combined_sentiment(x['text_sentiment'], x['point']), axis=1)
    return group

# 결과를 카페 아이디별로 그룹화하여 감성 레이블링
result_df = df_cafe.groupby('cafe_id').apply(apply_combined_sentiment)

# 카페 아이디별 감성 분석 결과 요약
cafe_summary = result_df.groupby('cafe_id')['combined_sentiment'].mean().reset_index()
cafe_summary.columns = ['cafe_id', 'average_sentiment']

# 퍼센티지로 변환 및 소수점 한 자리로 제한
cafe_summary['average_sentiment'] = (cafe_summary['average_sentiment'] * 100).round(1)
```

```
# 4. 감성 분석 결과를 MySQL에 업데이트
conn = mysql.connector.connect(
    host='192.168.0.26',
    user='user',
    password='mysql',
    database='main_project'
)
cursor = conn.cursor()

# 테이블 업데이트
update_query = """
UPDATE cafe
SET review_point = %s
WHERE cafe_id = %s
"""

for _, row in cafe_summary.iterrows():
    cursor.execute(update_query, (row['average_sentiment'], row['cafe_id']))

# 변경 사항 저장
conn.commit()

# 커넥션 종료
cursor.close()
conn.close()
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

3. 카페 검색



한글은 형태의 변형이 매우 복잡한 언어로써 특히 복합어, 합성어 등이 많아 하나의 단어도 여러 어간으로 분리해야 하는 경우가 많아 한글을 형태소 분석을 하려면 반드시 한글 형태소 사전이 필요

프로젝트에 엘라스틱서치의 **노리 (nori) 한글 형태소 분석기**를 적용하여 검색의 질적 수준을 높였음

회원, 카페, 회원활동, 선호유형 테이블을 융합하여 하나의 뷰로 생성 및 엘라스틱서치에 제공하여 검색시 다양한 정보를 제공 할 수 있도록 함

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

4. 추천 코드 예시 (Python)

```
from opensearchpy import OpenSearch
import json
import sys

# UTF-8 인코딩 설정
sys.stdout.reconfigure(encoding='utf-8')

# OpenSearch 클라이언트 설정
try:
    es = OpenSearch("http://192.168.0.26:9200")
except Exception as e:
    print(f"Error connecting to OpenSearch: {e}")
    exit(1)

# Step 1: 리뷰 수 집계 쿼리 설정 (리뷰가 있는 모든 문서 대상으로 집계)
review_query = {
    "size": 0,
    "query": {
        "exists": {
            "field": "review"
        }
    },
    "aggs": {
        "top_cafes": {
            "terms": {
                "field": "cafe_id",
                "size": 20,
                "order": {"_count": "desc"}
            }
        }
    }
}

# Step 2: 리뷰 수 집계 쿼리 실행
review_response = es.search(index="vw_member_act_index", body=review_query)

# Step 3: cafe_id와 review_count 추출
cafe_reviews = {bucket['key']: bucket['_doc_count']} for bucket in review_response['aggregations']['top_cafes']['buckets']
cafe_ids = list(cafe_reviews.keys())
```

```
# Step 4: vw_cafe_index에서 cafe_id, cafe_name, image_url 조회 쿼리 설정
cafe_query = {
    "query": {
        "terms": {
            "cafe_id": cafe_ids
        }
    },
    "source": ["cafe_id", "cafe_name", "image_url"],
    "size": 50
}

# Step 5: 카페 정보 쿼리 실행
cafe_response = es.search(index="vw_cafe_index", body=cafe_query)

# Step 6: 카페 정보를 리뷰 수와 결합하여 최종 결과 생성
final_results = []
for hit in cafe_response['hits']['hits']:
    cafe_id = hit['_source'].get('cafe_id', 'N/A')
    final_results.append({
        "cafe_id": cafe_id,
        "cafe_name": hit['_source'].get('cafe_name', 'N/A'),
        "image_url": hit['_source'].get('image_url', 'N/A'),
        "review_count": cafe_reviews.get(cafe_id, 0) # review_count 추가
    })

# Step 7: review_count (리뷰 수) 기준으로 재정렬
final_results.sort(key=lambda x: x['review_count'], reverse=True)

# Step 8: JSON 형식으로 결과를 문자열로 변환하여 출력
print(json.dumps(final_results, ensure_ascii=False, indent=2))
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

4. 추천 코드 예시 (JAVA)

```
//파이썬 파일내용을 입력받아서 실행시킨후 결과를 jsonArray로 반환하는 코드
@Service
public class PythonRead {
    public JSONArray pythonRead(String pyName,String argument,String argument2) {
        try {
            Resource resource = new ClassPathResource("scripts/"+pyName);
            InputStream inputStream = resource.getInputStream();

            BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream, StandardCharsets.UTF_8));
            StringBuilder scriptContent = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                scriptContent.append(line).append(System.lineSeparator());
            }

            File tempScript = File.createTempFile("temp_script", ".py");
            try (BufferedWriter writer = new BufferedWriter(new FileWriter(tempScript))) {
                writer.write(scriptContent.toString());
            }

            System.out.println(argument+" "+argument2);
            ProcessBuilder pb = new ProcessBuilder("python",tempScript.getAbsolutePath(),argument,argument2);
            Process p = pb.start();

            BufferedReader in = new BufferedReader(new InputStreamReader(p.getInputStream(), StandardCharsets.UTF_8));
            StringBuilder output = new StringBuilder();
            String line2;
            while ((line2 = in.readLine()) != null) {
                output.append(line2);
            }

            JSONArray jsonArray = new JSONArray(output.toString());

            return jsonArray;
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

PythonRead.java

```
@Controller
public class IndexController {

    @Autowired
    PyToCafeArr pyToCafeArr;

    //Index 접근용
    @RequestMapping("/index")
    public String connectIndex(Model model) {
        model.addAttribute("LRArrCDTO", pyToCafeArr.pyToCafeArr("cafe_likes_top20.py", "", ""));
        model.addAttribute("RRArrCDTO", pyToCafeArr.pyToCafeArr("cafe_reviews_top20.py", "", ""));
        return "Main";
    }

    //좋아요많은는 더보기
    @RequestMapping("likeCafeMore.do")
    public String likeCafeMore(Model model) {
        model.addAttribute("LRArrCDTO", pyToCafeArr.pyToCafeArr("cafe_likes_top20.py", "", ""));
        return "recommand/Like";
    }

    //리뷰많은는 더보기
    @RequestMapping("reviewsCafeMore.do")
    public String reviewsCafeMore(Model model) {
        model.addAttribute("RRArrCDTO", pyToCafeArr.pyToCafeArr("cafe_reviews_top20.py", "", ""));
        return "recommand/Review";
    }
}
```

IndexController.java

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

5. MVC2, 제어의 역전(IoC)



스프링 프레임워크를 사용해 MVC2 모델의 도메인형 디렉토리 구조로 웹페이지를 구현함. 기능별로 패키지를 나누어 컨트롤러, DTO, 서비스, 매퍼를 각각 구현하고, 공통적으로 사용하는 요소는 common 패키지로 관리함.

또한, index 패키지를 통해 메인 페이지 컨트롤러를 생성함.

스프링의 의존성 주입(DI)과 제어의 역전(IoC)을 활용해 어노테이션 방식으로 빈을 생성, 코드 재사용성과 유지보수를 강화함.

다만, 도메인형 구조에서는 위치가 애매한 클래스들이 많아지는 문제가 있었음.

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

5. MVC2, 제어의 역전(IOC)

```
@Controller
public class GoodController {

    @Autowired
    GoodService goodService;
    @Autowired
    CafeDetail cafeDetail;
    @Autowired
    GetReview getReview;

    //마이페이지 좋아요
    @RequestMapping(value = "myGood.do", method = RequestMethod.GET)
    public ModelAndView getLikedCafes(@RequestParam int member_id, @RequestParam(defaultValue = "1") int page) {
        int pageSize = 2; // 페이지당 카페 수
        int totalCafes = goodService.getLikedCafesCount(member_id); // 전체 카페 수
        List<CafeDTO> cafes = goodService.getLikedCafes(member_id, page, pageSize); // 페이징 처리된 카페 목록
        System.out.println("=====eee===== "+page);
        ModelAndView mav = new ModelAndView("mypage/mypage_like");
        mav.addObject("cafes", cafes);
        mav.addObject("currentPage", page);
        mav.addObject("totalPages", (int) Math.ceil((double) totalCafes / pageSize));

        return mav;
    }

    @RequestMapping("goMyReview.do")
    public String goMyReview(@RequestParam("member_id") String memberId, Model model) {
        model.addAttribute("MRArrCDTO", goodService.getMyReview(Integer.parseInt(memberId)));
        return "mypage/mypage_review";
    }
}
```

GoodController.java

```
@RequestMapping("goMyReview.do")
public String goMyReview(@RequestParam("member_id") String memberId, Model model) {
    model.addAttribute("MRArrCDTO", goodService.getMyReview(Integer.parseInt(memberId)));
    return "mypage/mypage_review";
}

//마이페이지 좋아요 취소
@RequestMapping(value="removeCafeLike.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8")
@ResponseBody
public String removeLike(@RequestParam("cafe_id") String cafe_id, HttpSession session) {
    System.out.println("취소----->"+cafe_id);
    MemberDTO id = (MemberDTO)session.getAttribute("user");
    if(id != null && cafe_id != null) {
        int member_id = id.getMember_id();
        goodService.removeLike(String.valueOf(member_id), cafe_id);
        goodService.deleteNull(String.valueOf(member_id), cafe_id);
        System.out.println("멤버아이디----->"+member_id);

        return String.format("{\\"success\\":true, \\"member_id\\": \"%d\\"}", member_id);
    } else {
        return "{\\"success\\":false}";
    }
}

//상세페이지 좋아요
@RequestMapping(value="addGood.do", method=RequestMethod.POST)
@ResponseBody
public String addGood(@RequestParam("memberId") String memberId,
    @RequestParam("cafeId") String cafeId) {
    if(memberId.equals("")) {
        return "needlogin";
    }
    String findGoodResult = goodService.findGood(memberId, cafeId);
    if(findGoodResult != null && findGoodResult.equals("Y")) {
        return "already";
    }
    String addUpdate = goodService.add_OR_updateGood(memberId, cafeId);
    if(addUpdate != null) {
        goodService.updateGood(memberId, cafeId);
    } else {
        goodService.addGood(memberId, cafeId);
    }
    return "done";
}
```

9. 프로젝트 수행 경과

- 데이터 수집 가공 및 제공

5. MVC2, 제어의 역전(IOC)

```
@Service
public class GoodService {

    @Autowired
    GoodMapper goodMapper;

    //데이터필요한 방안
    private Map<String, Object> makeMap(String userid, String cafeid, Float rating, String review, String cafe_mood) {
        Map<String, Object> params = new HashMap<>();
        params.put("userid", Integer.parseInt(userid));
        params.put("cafeid", Integer.parseInt(cafeid));
        if(rating!=null) {
            params.put("rating", rating);
        }
        if(review!=null) {
            params.put("review", review);
        }
        if(cafe_mood!=null) {
            params.put("cafe_mood", cafe_mood);
        }
        return params;
    }

    // 데이터베이스에서 카페 정보를 조회
    public List<CafeDTO> getLikedCafes(int member_id, int page, int pageSize) {
        Map<String, Object> params = new HashMap<>();
        params.put("member_id", member_id);
        params.put("limit", pageSize);
        params.put("offset", (page - 1) * pageSize);
        return goodMapper.getLikedCafes(params);
    }

    public int getLikedCafesCount(int member_id) {
        return goodMapper.getLikedCafesCount(member_id);
    }

    public List<MyReviewDTO> getMyReview(int userid) {
        return goodMapper.getMyReview(userid);
    }
}
```

```
public void removeLike(String member_id, String cafe_id) {
    goodMapper.removeLike(makeMap(member_id, cafe_id, rating:null, review:null, cafe_mood:null));
}

public void deleteNull(String member_id, String cafe_id) {
    goodMapper.deleteNull(makeMap(member_id, cafe_id, rating:null, review:null, cafe_mood:null));
}

public String findGood(String userid, String cafeid) {
    return goodMapper.findGood(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public String add_OR_updateGood(String userid, String cafeid) {
    return goodMapper.addUpdateGood(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public void addGood(String userid, String cafeid) {
    goodMapper.addGood(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public void updateGood(String userid, String cafeid) {
    goodMapper.updateGood(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public String findReview(String userid, String cafeid) {
    return goodMapper.findReview(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public String add_OR_updateReview(String userid, String cafeid) {
    return goodMapper.addUpdateReview(makeMap(userid, cafeid, rating:null, review:null, cafe_mood:null));
}

public void addReview(String userid, String cafeid, float rating, String review, String cafe_mood) {
    goodMapper.addReview(makeMap(userid, cafeid, rating, review, cafe_mood));
}

public void updateReview(String userid, String cafeid, float rating, String review, String cafe_mood) {
    goodMapper.updateReview(makeMap(userid, cafeid, rating, review, cafe_mood));
}
```

12. 평가

사전 기획 관점에서 프로젝트 결과물에 대한 완성도 평가:

이번 프로젝트는 초기 기획 단계에서 목표와 범위를 비교적 유연하게 설정하였습니다. 여러 기능과 목표를 구상했으나, 우선적으로 필수적인 부분부터 구현한 후 최종 목표를 확정하자는 접근 방식을 취했습니다. 이 과정에서 모든 기능을 구현하지는 못했지만, 필수적인 기능들은 완성도 높게 구현되었습니다. 이러한 유연한 기획 덕분에 핵심 목표에 집중할 수 있었고, 최종적으로 프로젝트의 목적을 달성했다고 평가할 수 있습니다.

12. 평가

추후 개선점, 보완할 점

1. 기능 확장

초기 기획 단계에서 구현하지 못한 기능들을 추가 개발하여 더 많은 기능을 사용자에게 제공할 수 있습니다. 이를 통해 사용자 요구를 충족시키고, 서비스의 가치를 높일 수 있을 것입니다.

2. 데이터 품질 향상

개인 정보가 필요하여 다수의 데이터를 가공하여 작업했습니다. 추후에는 보다 고품질의 데이터를 수집하고, 이를 기반으로 정밀한 분석을 통해 서비스의 신뢰도를 더욱 높일 수 있을 것입니다.

12. 평가

추후 개선점, 보완할 점

3. UI/UX

사용자 친화적인 인터페이스를 구현하는 것이 중요합니다. 현재의 직관적이지 않거나 복잡한 부분을 간소화하여 사용자 경험을 개선하고, 접근성을 강화할 필요가 있습니다.

4. 성능 최적화

다양한 어플리케이션이 사용됨에 따라 데이터 처리 속도가 최적화되지 않은 부분이 있습니다. 코드 구조를 개선하여 성능을 최적화함으로써 사용자에게 더 빠르고 효율적인 서비스를 제공할 수 있습니다.

12. 평가 - 이상현

잘한 부분, 아쉬운 점

기획부터 설계까지 모든 팀원이 빠짐없이 의견을 제시하며 협력해 프로젝트를 완성한 점이 잘한 부분이라 생각합니다. 구현 단계에서도 각자가 겪은 어려움 속에서 포기하지 않고 서로를 도우며 함께 해결해 나간 것이 매우 긍정적이었습니다. 하지만 시간이 조금 더 있었다면, 더 많은 기능과 높은 완성도를 구현할 수 있었을 것이라는 아쉬움이 남습니다.

느낀점이나 경험한 성과

팀장으로서 프로젝트의 방향성을 제시하고 의사결정을 하며 전체 진행 상황을 관리하는 것은 저에게 큰 도전이었습니다. 그러나 팀원들과 각 파트의 리더들이 열심히 협력해준 덕분에 큰 문제 없이 프로젝트를 완성할 수 있었습니다. 이번 경험을 통해, 앞으로 리더의 역할을 맡게 된다면 이 경험이 큰 도움이 될 것이라고 믿습니다.

12. 평가 - 차지민

잘한 부분, 아쉬운 점

기획 단계에서 워크플로우를 계획하고 그대로 구현에 성공한 점, 구상한 모든 기능을 구현한 점, 그리고 내가 담당한 부분에서 완전한 의존성 역전을 구현하고 유효성 검사 등 세부 사항까지 신경 쓴 점이 잘한 부분입니다. 하지만 스프링 프레임워크를 사용하면서 엘라스틱 서치와의 연동에 아쉬움이 있었고, 메인 로딩이나 검색 마커 등의 성능 문제, 도메인형 패키지 구조의 애매함으로 공용 패키지가 많아졌다는 점, DB 설계에서의 문제를 개선하지 못한 점은 아쉬운 부분입니다.

12. 평가 - 차지민

느낀점이나 경험한 성과

처음에는 스프링 프레임워크가 어려웠지만, 실제로 사용해보니 큰 어려움 없이 활용할 수 있었습니다. 스프링 프레임워크를 배우면서 스프링 부트에 대한 호기심이 생겼고, EL과 JSTL의 숙련도가 오를수록 JSP보다 훨씬 간편하고 효율적이라는 점을 깨달았습니다. 또한, 마이바티스의 활용도를 알게 되었고, 모호했던 스프링 프레임워크와 EL/JSTL에 대한 이해가 명확해졌으며, 이를 통해 자신감을 얻게 되었습니다. 반년 동안 배운 여러 기술 스택들이 어떻게 융합되어 작동하는지 개념이 잡힌 것도 큰 성과입니다.

12. 평가 - 김은규

잘한 부분, 아쉬운 점

스프링의 흐름을 조금이나마 이해할 수 있었으며, 경력자인 지민님과 백엔드 작업을 함께하면서 내가 놓친 부분을 새로운 시각으로 제시해주어 큰 도움을 받았습니다. 특히 팀 간의 소통이 원활하게 이루어진 점이 매우 긍정적이었습니다.

느낀점이나 경험한 성과

첫 번째 프로젝트보다 더 디테일한 코딩을 할 수 있었고, 더 많은 인원이 각자 구역을 나누어 작업한 것이 재미있었습니다. 프로젝트를 통해 스프링을 더 쉽게 이해하게 되었으며, 두 번의 프로젝트 모두 끝까지 포기하지 않고 꾸준히 도전한 점이 큰 성과였습니다. 에러가 발생해도 이를 해결하는 과정에서 부담감이 줄어들었고, 코딩에 더욱 흥미를 느끼게 되었습니다.

12. 평가 - 백찬혁

잘한 부분, 아쉬운 점

프론트엔드 팀의 전체 회의가 잘 진행되어 색상 결정 및 파트 나누기 등의 과정이 원활하게 이루어졌습니다. 그러나 내가 맡은 파트를 완수하기 위해 시간 분배를 더 고려하지 못한 점은 아쉬움으로 남습니다.

느낀점이나 경험한 성과

시간 분배와 백엔드 진행 속도에 맞추기 위해 우선순위가 높은 페이지를 먼저 구현하는 것이 중요하다는 것을 깨달았습니다.

12. 평가 - 이경희

잘한 부분, 아쉬운 점

잘한 점: 교육 과정에서 습득한 기술들을 프로젝트에 많은 부분 적용한 점 팀원 간 협업도 원활하게 이루어짐.

개선할 점: 습득한 많은 기술을 적용하긴 했지만, 기능 구현의 완성도는 조금 부족.

느낀점이나 경험한 성과

- 프로젝트를 수행하면서 서로 부족한 부분을 보완하고 협력하는 팀워크가 돋보임.
- 각자의 역할 외에도 자발적으로 문제를 해결하려는 태도가 프로젝트 뿐이 아닌 이번 교육과정에서 얻은 큰 수확이라고 여기짐.
- 각 개인의 성장에도 큰 도움이 되었다고 생각됨.

12. 평가 - 이운형

잘한 부분, 아쉬운 점

팀원들이 짧은 시간 동안 서로의 부족한 부분을 채워주며 프로젝트를 성공적으로 완료했습니다. 하지만 각 페이지의 규격과 글씨 크기, 위치를 일관되게 유지하지 못한 점은 아쉬움으로 남습니다.

느낀점이나 경험한 성과

프론트엔드 니를 만들면서 좀 더 깔끔하고 아름답게 디자인할 수 있었을 것 같았지만, 그렇게 하지 못해 아쉬움이 남습니다. 그럼에도 불구하고 프론트 개발 실력이 조금 향상된 점은 긍정적입니다. 앞으로는 미흡한 백엔드와 데이터베이스 부분을 더 공부하여 혼자서 프로젝트를 완성할 수 있는 실력을 기르고 싶습니다.

12. 평가 - 이주연

잘한 부분, 아쉬운 점

팀원들은 각자 맡은 역할을 충실히 수행하며 협업을 통해 프로젝트를 성공적으로 완수할 수 있었습니다. 역할 분담이 효과적으로 이루어졌으며, 각자의 책임 영역을 잘 소화해낸 점이 돋보였습니다.

느낀점이나 경험한 성과

프로젝트를 통해 두 번의 프론트엔드 경험을 해보며, 처음에는 막막했지만 본 프로젝트에서는 보다 수월하게 프론트엔드를 다룰 수 있었습니다. 여전히 모르는 부분이 많아 지속적인 학습의 필요성을 느꼈으며, 앞으로는 백엔드 공부도 병행하여 더 다양한 기술 역량을 쌓고 싶습니다.

프로젝트 시연 영상

