






초급 개발자 **김도빈** 포트폴리오

— All 웹툰 —

Contact

 zkffhtm6523@naver.com

 010-2532-6523

 zkffhtm6523@github.com

1 All 웹툰

웹 크롤링을 활용한 한국의 웹툰 추천 서비스 사이트 구현

 allwebtoon.xyz

 **GitHub**

- 프로젝트 소개
- 기능 요약
- ERD
- SNS API
- Web Crawling
- Ajax 비동기 통신
- Mahout 추천 알고리즘
- NCP(Naver Cloud Platform)
- 도메인 연결(Gabia)

- 1 All 웹툰
- 1 프로젝트 소개
- 2 기능 요약
- 3 ERD
- 4 SNS API
- 5 Crawling
- 6 Ajax
- 7 Mahout
- 8 NCP
- 9 도메인 연결

- 프로젝트 소개

개발 목적	웹 크롤링을 활용한 한국의 웹툰 추천 서비스 사이트 구현
개발 기간	2020.08.03 ~ 2020.10.15
사용 도구	Eclipse(개발 도구), Mysql(데이터베이스), git(협업 도구)
개발 인원	3명(김도빈, 석규환, 정혜선)
담당 업무	SNS API를 이용한 로그인 서비스 환경 구축
	Web Crawling을 이용한 웹툰 정보 수집(Daum, Toptoon)
	Apache Mahout을 이용한 추천 알고리즘 시스템
	Ajax(비동기 통신)
	HTML, CSS, Javascript를 활용한 View Page 디자인
	Web Hosting & DNS

목차

1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

5 Crawling

6 Ajax

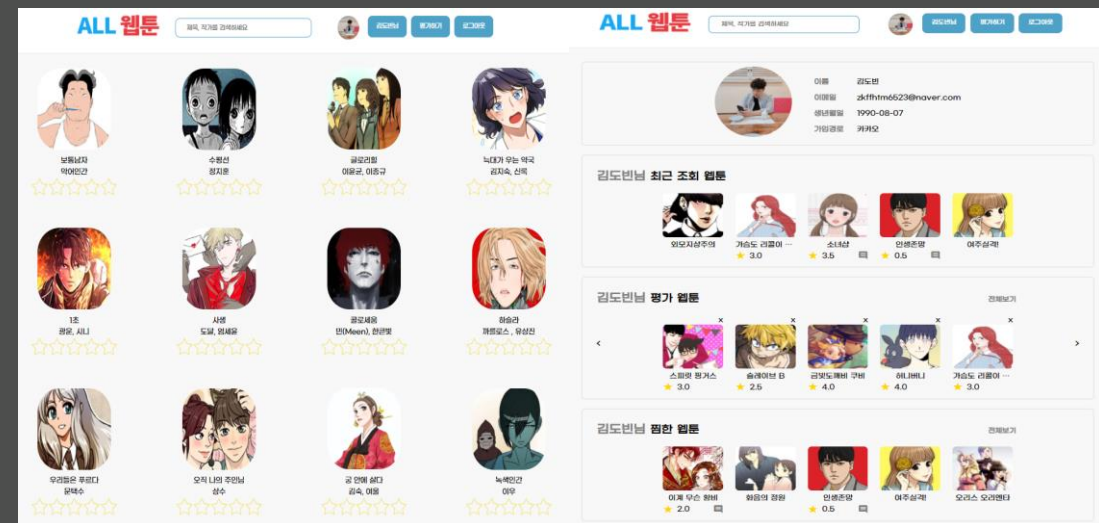
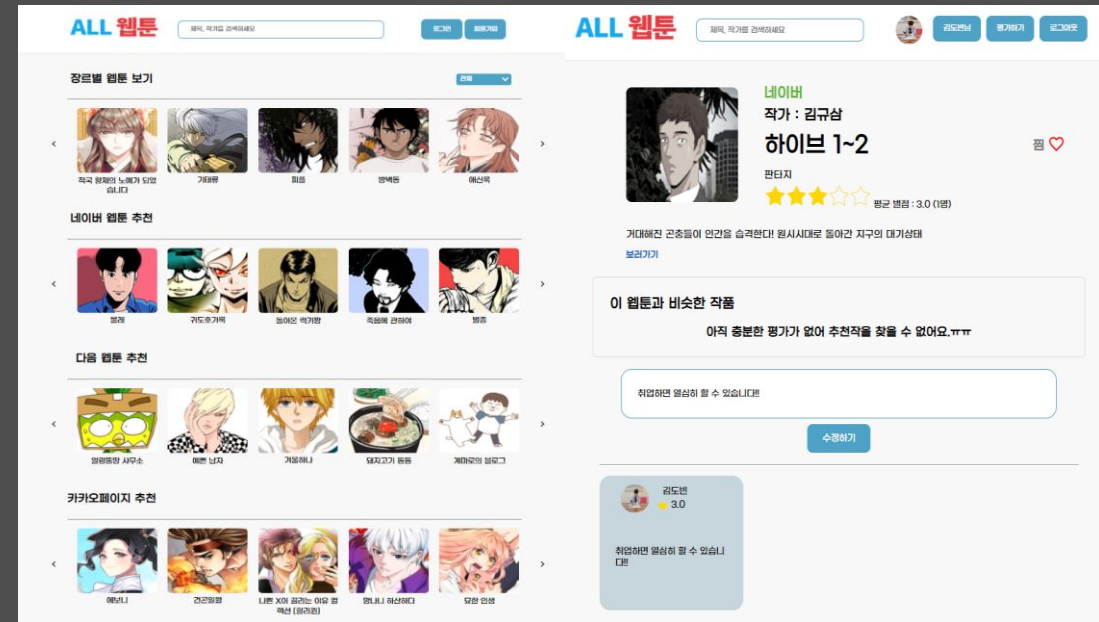
7 Mahout

8 NCP

9 도메인 연결

- 기능 요약

- 크롤링 활용 후 웹툰 플랫폼에서 만화 정보 수집
- 왓차피디아의 시스템을 모티브로 프로젝트 진행
 - 웹툰 검색 및 정보 파악
 - 유저들의 평점을 기반으로 한 추천 시스템
- 다양한 기술적 시도가 녹여져 있는 프로젝트
 - Mahout, Python 추천 시스템
 - Ajax 비동기 통신 및 JSON 활용
 - NCP, Filezilla를 활용한 Website Hosting
 - 가비아를 통한 도메인 연결
 - SNS API를 활용한 Login 서비스 제공
- JSP 및 Servlet을 이용한 MVC패턴



1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

5 Crawling

6 Ajax

7 Mahout

8 NCP

9 도메인 연결

- 기능 요약

ALL 웹툰 기능 설명 동영상입니다.
(클릭 시 Youtube로 동영상 이동)

ALL 웹툰 기획서입니다.
(클릭 시 Github로 이동)

All webtoon
기능 설명

ALL 웹툰

김도빈 정혜선 석규환

목차

- ERD(Entity-Relation Diagram)

1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

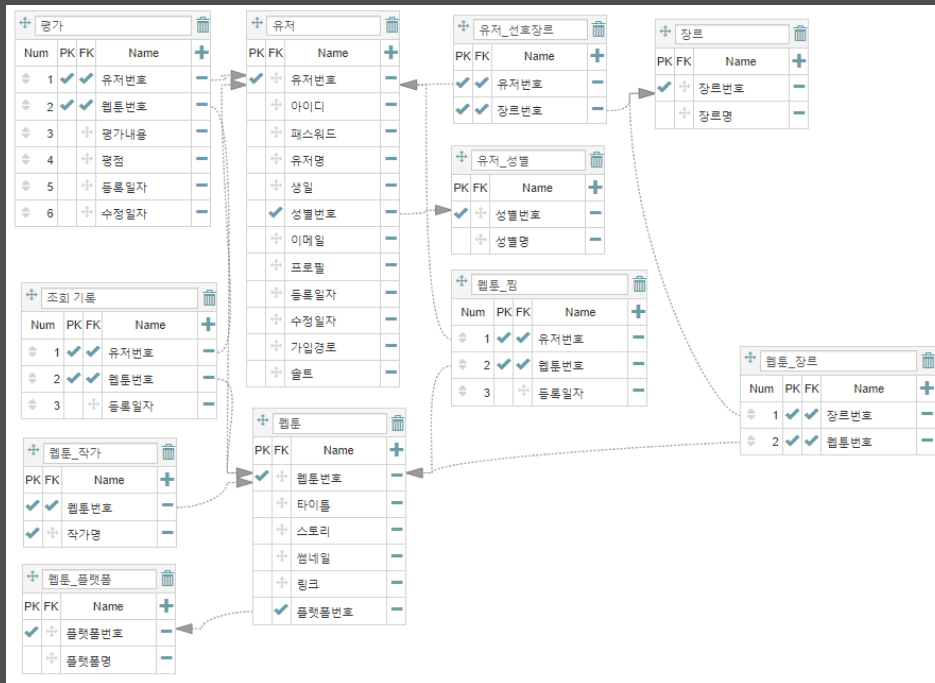
5 Crawling

6 Ajax

7 Mahout

8 NCP

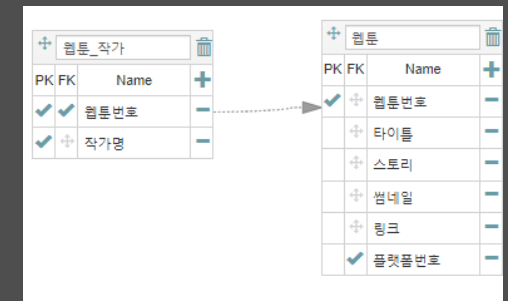
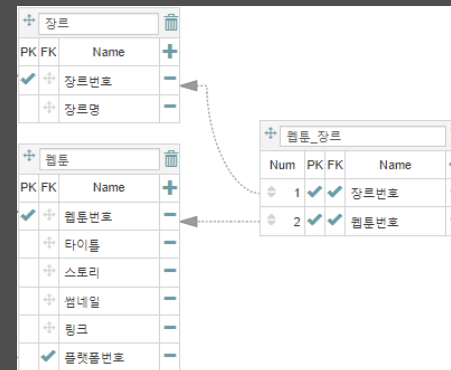
9 도메인 연결



논리 모델 형태

- 유저, 웹툰, 평가를 기본으로 모델 구성
- 작가, 선호장르는 중복으로 인한 **1정규화**
- 조회 기록, 찜은 데이터가 있다면 출력 및 최신순으로 **내림차순 정렬**
- 장르, 성별은 관리용으로 사용

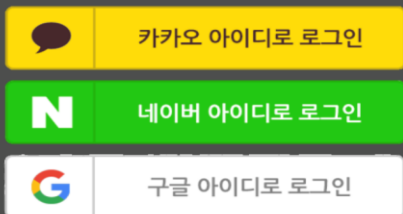
- 작가, 선호장르는 2개 이상의 값을 가지는 경우가 많음
- 값 중복으로 인한 테이블 분리 및 **제1 정규화**



- 1 All 웹툰
- 1 프로젝트 소개
- 2 기능 요약
- 3 ERD
- 4 SNS API
- 5 Crawling
- 6 Ajax
- 7 Mahout
- 8 NCP
- 9 도메인 연결

- SNS API

: API를 활용한 사용자들에게 로그인 및 회원가입 편의성 제공



각 플랫폼 별 버튼 클릭 시
해당 플랫폼 **Login 화면** 이동

```
String access_token = getAccessToken(request.getParameter("code"));
UserInfo userInfo;
try {
    userInfo = getUserInfo(access_token);
    int result = UserDao.selectSNSUser(userInfo);
    if(result == 0) {
        request.setAttribute("userInfo", userInfo);
        ViewResolver.accessForward("join", request, response);
        return;
    }
}
```

Code 포함 URI 전송

Token을 플랫폼으로부터 획득

Parsing 후 JSON 형태의 정보

Redirect URI	
Redirect URI	http://allwebtoon.xyz/kakaoAPI http://101.101.219.238/kakaoAPI http://localhost:8089/kakaoAPI
네이버아이디로로그인 Callback URL (최대 5개)	
	<input type="text" value="http://localhost:8089/naverAPI"/> - <input type="text" value="http://101.101.219.238/8080/naverAPI"/> - <input type="text" value="http://allwebtoon.xyz/naverAPI"/> +
승인된 리디렉션 URI ⓘ 웹 서버의 요청에 사용	
URI <input type="text" value="http://localhost:8089/googleAPI"/> <input type="text" value="http://allwebtoon.xyz/googleAPI"/>	
<input type="button" value="+ URI 추가"/>	

Login 성공 시
Redirect URI로 Code 전달

```
param.setU_id(user_id);
param.setU_password(user_id);
param.setU_joinPath(z);

if(!"".equals(email)) {param.setU_email(email);}
if(!"".equals(nickname)) {param.setU_name(nickname);}
if(!"".equals(profile_image)) {
    param.setU_profile(profile_image);
    param.setChkProfile(param.getU_profile().substring(0, 4));
}
if(!"".equals(gender)) {
    param.setGender_name(gender.equals("male") ? "남성" : "여성");
}
```

Key-Value값으로 전달된 정보
JSON->String으로 Parsing 후
VO에 저장

각 플랫폼별 **JoinPath**를
부여함으로써 가입 경로 파악

목차

1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

5 Crawling

6 Ajax

7 Mahout

8 NCP

9 도메인 연결

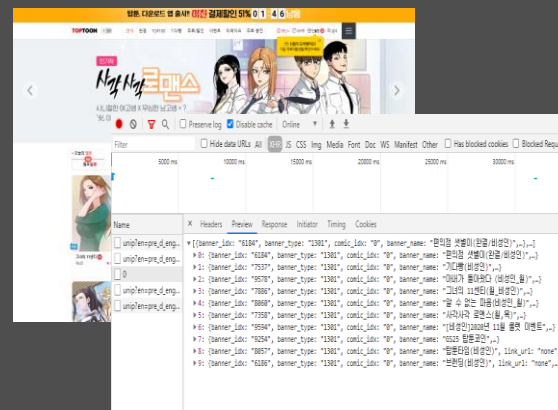
- Crawling

: Web Crawling를 활용한 웹툰 정보를 Database에 정보 삽입

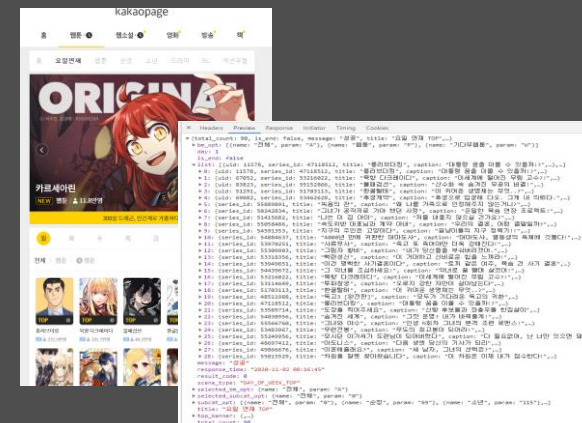


국내 대형 웹툰 플랫폼 5개 선정

웹툰 사이트에서 정보 제공 URI 파악
JSON의 규칙성 파악 및 필요 정보 파악



탑툰



카카오페이지

목차

- Crawling

: Web Crawling를 활용한 웹툰 정보를 Database에 정보 삽입

```
// 4-1. 다음 웹툰 연재작 DB 담기(카카오와 중복 제거되서 총 161개 중 126개 삽입)
Daum.getDaum(list);
for (int i = 0; i < list.size(); i++) {
    CrawWebtoonDAO.insWebtoonList(list.get(i));
}
// 4-2. 다음 웹툰 연재작 DB 담기(카카오와 중복 제거되서 총 161개 중 126개 삽입)
Daum.getCompleteDaum(list);
for (int i = 0; i < list.size(); i++) {
    CrawWebtoonDAO.insWebtoonList(list.get(i));
}
```

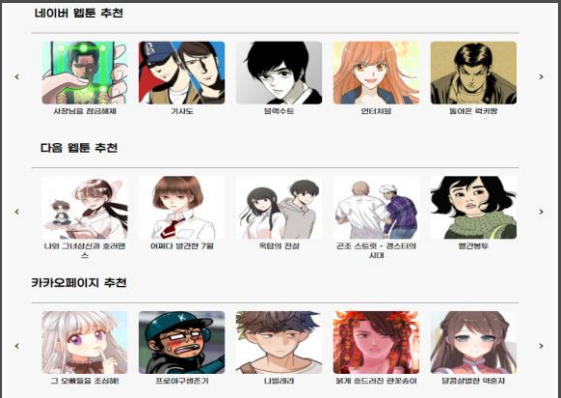
JSON 데이터 Parsing 후 list 데이터 담기

List의 배열을 SQL로 Database에 Insert

웹툰 플랫폼별 분류 및 데이터 저장

#	이름	데이터 유형
1	w_no	INT
2	w_title	VARCHAR
3	w_story	VARCHAR
4	w_thumbnail	VARCHAR
5	w_link	VARCHAR
6	plat_no	INT

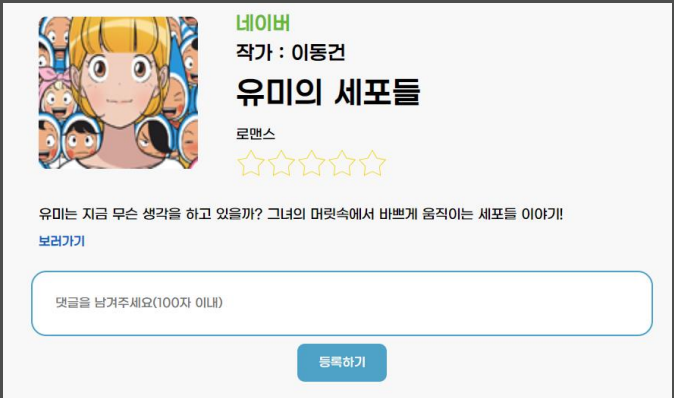
plat_no	plat_name
1	네이버
2	다음
3	카카오
4	레진코믹스
5	타툰



플랫폼별 웹툰 리스트



플랫폼 검색 결과



웹툰 상세화면 플랫폼 표시

- 1 All 웹툰
- 7 프로젝트 소개
- 2 기능 요약
- 3 ERD
- 4 SNS API
- 5 Crawling
- 6 Ajax
- 7 Mahout
- 8 NCP
- 9 도메인 연결

목차

1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

5 Crawling

6 Ajax

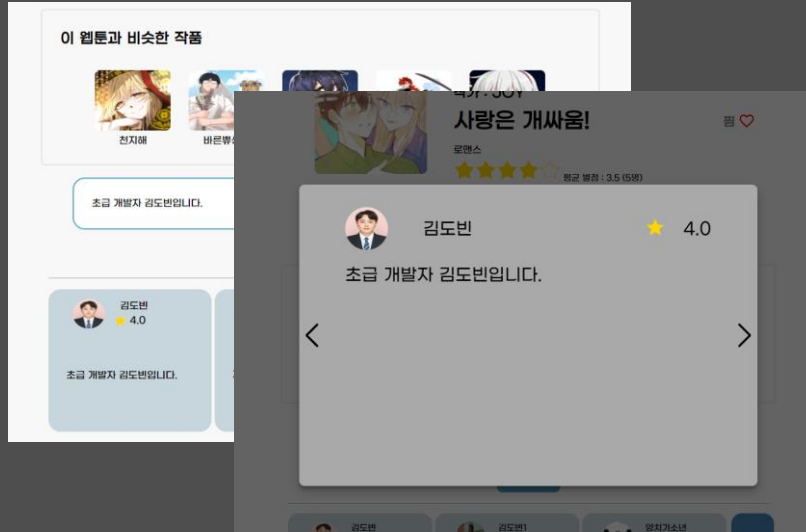
7 Mahout

8 NCP

9 도메인 연결

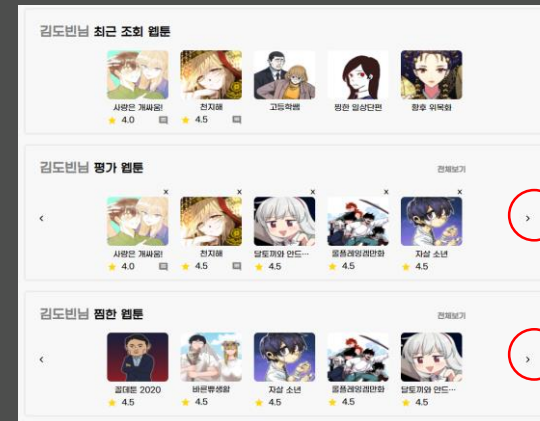
- Ajax(비동기 통신)

: 비동기 통신을 활용한 페이지 새로 고침 없이 다양한 기능 제공



Modal 화면 출력 시 **Ajax**를 활용한
평가 유저 정보 가져오기

버튼 클릭 시 **Ajax**를 활용한
다음 Index의 값 가져오기



버튼 클릭 전



버튼 클릭 후

- Mahout 추천 알고리즘

: 사용자 평점 기반으로 유사한 웹툰을 사용자에게 추천하는 시스템

1 All 웹툰

1 프로젝트 소개

2 기능 요약

3 ERD

4 SNS API

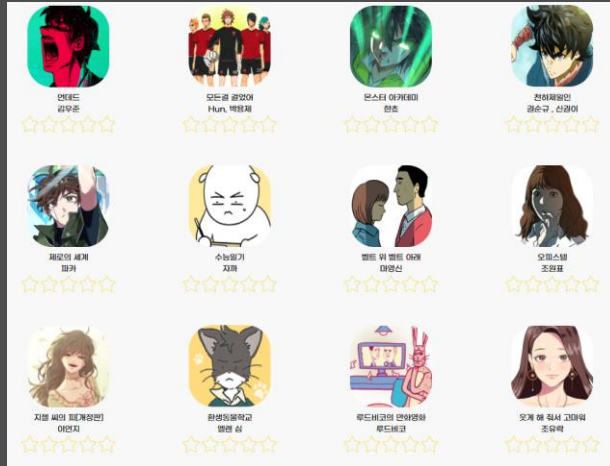
5 Crawling

6 Ajax

7 Mahout

8 NCP

9 도메인 연결



```
Integer category = null;
WebtoonCmtDomain tempVo = null;

for (int i = 0; i < tempList.size(); i++) {
    if (category == null || (category != null &&
        !category.equals(tempList.get(i).getU_no())) {
        //알의 u_no가 바뀔때만 생성
        category = tempList.get(i).getU_no();
        //영표
        WebtoonCmtDomain vo = new WebtoonCmtDomain();
        vo.setU_no(tempList.get(i).getU_no());

        List<WebtoonCmtVo> w_list = new ArrayList<WebtoonCmtVo>();
        vo.setW_list(w_list);
        tempVo = vo;
        webtoonList.add(vo);

        WebtoonCmtVo w_param = new WebtoonCmtVo();
        w_param.setW_no(tempList.get(i).getW_no());
        w_param.setC_rating(tempList.get(i).getC_rating());
        w_list.add(w_param);
        tempVo.setW_list(w_list);
        if (i == tempList.size()-1) {
            break;
        } else if (tempList.get(i).getU_no() != tempList.get(i+1).getU_no()) {
            continue;
        } else {
            i++;
        }
    }
    WebtoonCmtVo w_param1 = new WebtoonCmtVo();
    w_param1.setW_no(tempList.get(i).getW_no());
    w_param1.setC_rating(tempList.get(i).getC_rating());
    tempVo.getW_list().add(w_param1);
}
```

```
//2차원 배열 필요([u_no, [w_no][c_rating])
public static GenericDataModel parsingDataModel(List<WebtoonCmtDomain> webtoonList) {
    FastByIDMap<PreferenceArray> result = new FastByIDMap<PreferenceArray>();

    for (int i = 0; i < webtoonList.size(); i++) {
        System.out.println("-----u_no : "+webtoonList.get(i).getU_no()+"-----");
        List<Preference> prefsList = Lists.newArrayList();
        for (int j = 0; j < webtoonList.get(i).getW_list().size(); j++) {
            prefsList.add(new GenericPreference(
                webtoonList.get(i).getU_no(),
                webtoonList.get(i).getW_list().get(j).getW_no(),
                webtoonList.get(i).getW_list().get(j).getC_rating()
            ));
            System.out.print("w_no : "+webtoonList.get(i).getW_list().get(j).getW_no());
            System.out.println("/ c_rating : "+webtoonList.get(i).getW_list().get(j).getC_rating());
        }
        result.put(webtoonList.get(i).getU_no(), new GenericUserPreferenceArray(prefsList));
    }
    return new GenericDataModel(result);
}
```

평가하기를 통해서 유저 정보 수집

Key-Value 형태로 값 저장
(u_no, w_no, c_rating)

Datamodel 형태로 Parsing
(u_no, w_no, c_rating)

```
//피어슨 상관관계 : 사람들이 준 평점을 분석해 아이템 간의 유사도를 측정해줌. + 가중치
ItemSimilarity itemSimilarity = new PearsonCorrelationSimilarity(dataModel, Weighting.WEIGHTED);

//타니모토 계수 : 나와 평가한 작품의 교집합이 큰 사용자가 평가한 작품 중, 내가 보지 않은 것을 추천. (평가한 작품의 교집합을 찾지만 별점이 반영되지는 않을듯. ) : 0~1
//TanimotoCoefficientSimilarity itemSimilarity = new TanimotoCoefficientSimilarity(dataModel);

//로그우도 유사도 : 타니모토계수와 유사. 선호도 고려하지 않고 공통된 아이템 수 활용.
//ItemSimilarity itemSimilarity = new LogLikelihoodSimilarity(dataModel);

ItemBasedRecommender recommender = new GenericItemBasedRecommender(dataModel, itemSimilarity);
```

피어슨 상관관계 활용 유사도 산출
Datamodel + Similarity로 추천 리스트 생성



유저에게 추천 리스트 제공

목차

- 1 All 웹툰
- 1 프로젝트 소개
- 2 기능 요약
- 3 ERD
- 4 SNS API
- 5 Crawling
- 6 Ajax
- 7 Mahout
- 8 NCP
- 9 도메인 연결

- NCP(Naver Cloud Platform)

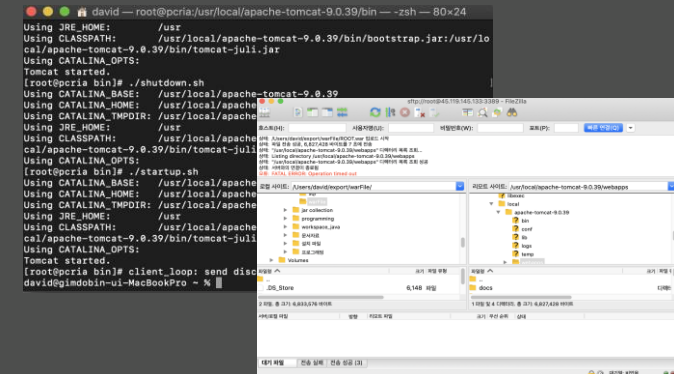
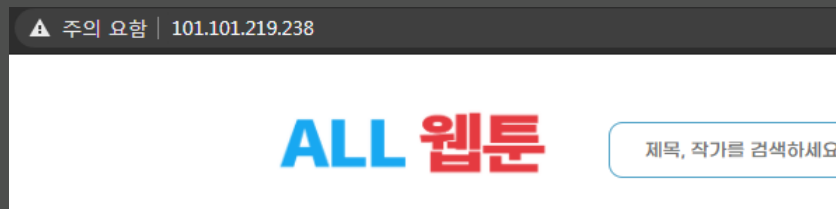
: NCP를 이용하여 웹 사이트 호스팅

<input type="checkbox"/> 서버 이름	서버 이미지 이름	서버 구성
<input type="checkbox"/> allwebtoon	mysql(5.7)-centos-7.2-64	[Micro] 1vCPU, 1GB Mem [g1]
Total Item: 1		

NCP를 통해서 서버 구축

DB접속 권한 설정 및 각종 라이브러리 서버에 설치
Filezilla로 WAR 파일 업로드

```
nel needs to be upgraded.  
$ sudo iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080  
$
```



포트번호 80을 포트번호 8080로 Redirect
: 포트번호 없이 도메인명으로 웹사이트 접근

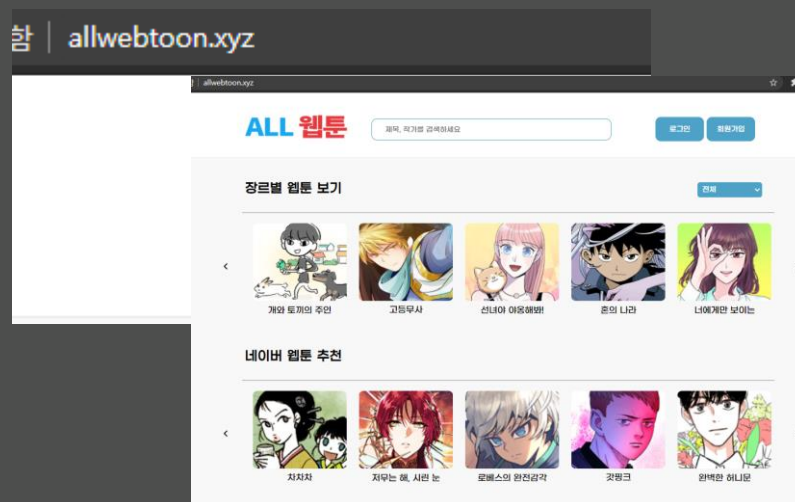
- 1 All 웹툰
- 1 프로젝트 소개
- 2 기능 요약
- 3 ERD
- 4 SNS API
- 5 Crawling
- 6 Ajax
- 7 Mahout
- 8 NCP
- 9 도메인 연결

- Gabia 도메인 연결

: 웹사이트의 접근성을 높이기 위해서 **도메인 구입 및 연동**

allwebtoon.shop	EVENT 500원 / 499,000원	선택
allwebtoon.site	EVENT 1,900원 / 499,000원	선택
allwebtoon.org	20,000원	선택
allwebtoon.me 몬테네그로	EVENT 6,000원 / 299,000원	선택
allwebtoon.한국 대한민국	20,000원	선택

www.allwebtoon.xyz / allwebtoon.xyz 연동



allwebtoon.com 도메인을 원했지만 사용중
allwebtoon.xyz 도메인 구매


DNS 관리		
allwebtoon.xyz		
* 레코드 개수 : 2개 * 최근 업데이트 : 2020-10-26 20:13:15 * 네임서버 : ns.gabia.co.kr		
DNS 설정 레코드 수정		
타입	호스트	
A	www	101.101.219.238
A	@	101.101.219.238


도메인명 확보를 통한 웹사이트 **접근성 향상**




감사합니다

Contact

 zkffhtm6523@naver.com

 010-2532-6523

 zkffhtm6523@github.com