2017. 2. 3.
7기 2단계 역량강화스터디 1월 평가

웹 전체를 아우르는 의견 공유 시스템

안재욱 윤승원 정현진 멘티

강대명 멘토님

목차

역량강화스
터디

서비스 소개

문제점

- 의견을 기능이 없는 사이트에서는 의견을 달 수 없다.
- 의견이 여러 플랫폼에 분산되어 있다

  의견을 달기 위해 매번 다른 플랫폼에 로그인해야 한다.

  플랫폼마다 다른 알림을 제공해서 통합된 관리가 어렵다.

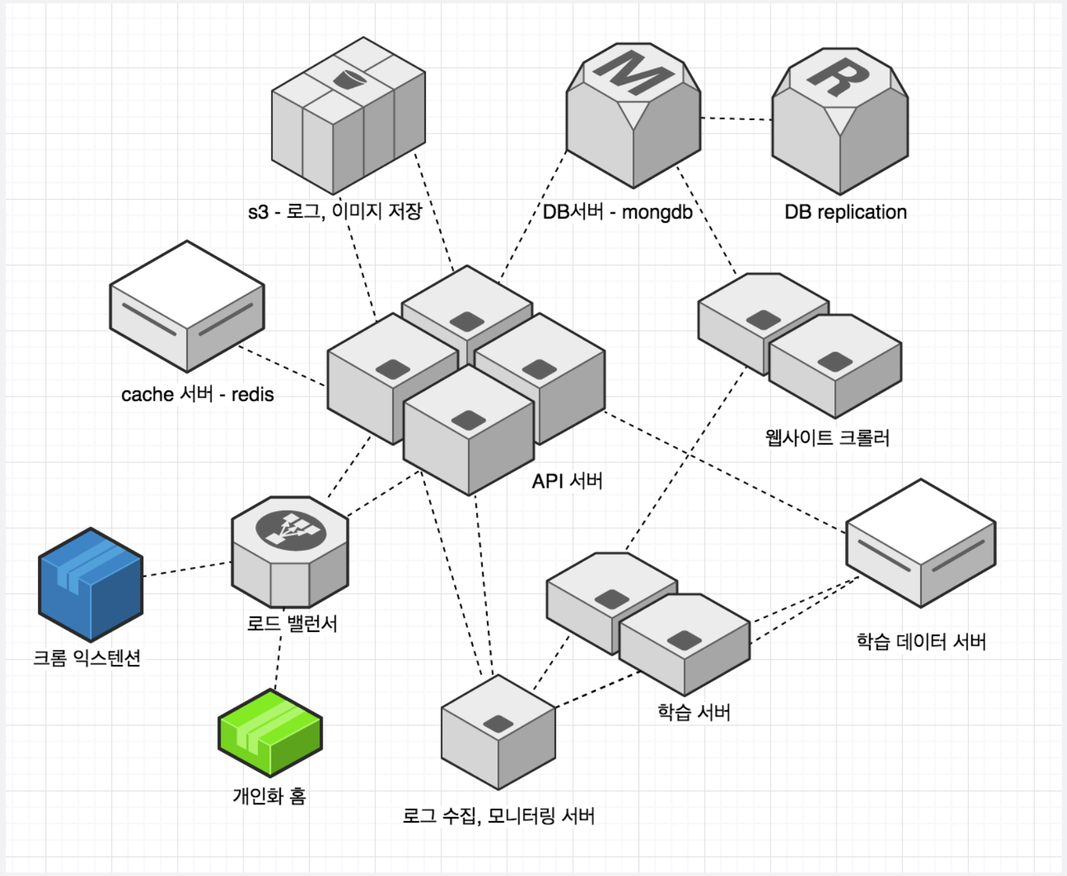
솔루션

크롬 익스텐션을 통해 모든 웹사이트에서 의견을 달 수 있다.

웹 사이트에서 단 모든 의견이 하나의 피드로 통합된다.

피드 추천을 통해 유저에게 도움이 되는 의견을 분류하여 제공한다.

전체
시스템



시스템 구성

- 크롬 익스텐션
- 개인화 홈
- API 서버
- Cache 서버
- DB 서버
- 학습 서버
- Amazon S3
- 웹사이트 크롤러
- 학습 데이터 서버
- 로깅, 모니터링 서버

역량강화스
터디

시나리오

01 대학교 1학년 문과 학생 A씨는 프로그래밍을 배워보려고 인터넷 검색을 했다. 하지만 ‘프로그래밍’으로 검색한 결과는 학원 광고와 사전적 정의뿐이라서 A씨에게 유용하지 않았다. 그래서 A씨는 사람들과 인터넷 페이지상에서 모르는 것이 있다면 바로바로 소통하고 싶다는 생각을 했다.

02 그런 A씨가 우리 서비스가 있다는 것을 알게 되고 주로 사용하는 브라우저인 크롬에서 바로 설치해, 프로그래밍 검색 결과 페이지에서 어떤 것부터 시작해야하는지 의견을 남긴 후, 뉴스를 보고 있었다.

03 잠시 후 A씨가 남긴 의견에 누군가가 답글을 달았다는 알림이 표시됐다. 새 탭을 여니 개인화 홈이 바로 노출되어 의견에 대한 답글을 확인하여 답글이 마음에 들어서 그 답글을 추천해줬다.

04 A씨가 답글을 확인하고 다시 돌아와 개인화 홈을 보니, 프로그래밍에 대한 또다른 피드들이 추천되어 있어서 A씨는 피드들을 둘러보며 자신이 모르던 여러가지 사실들을 알 수 있었다.

설계
개발할 기능 및 정책

필요한 기능

“개인화 홈(웹 서비스),
크롬 익스텐션(UI), API 및 DB 설계”

구체적인 정책 수립

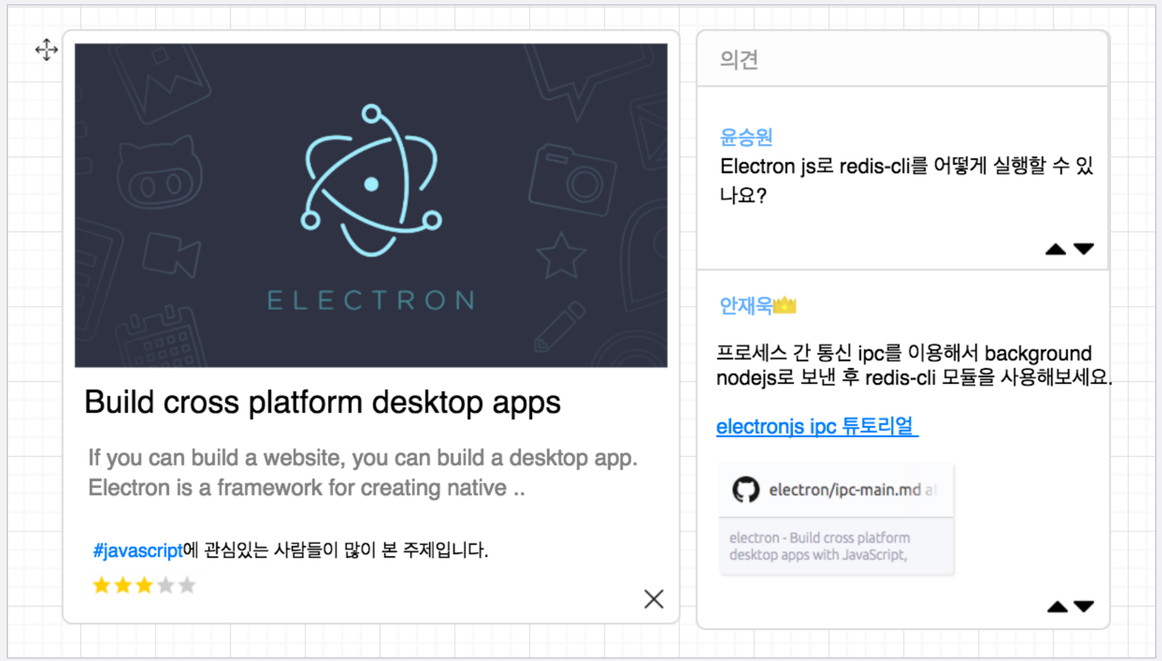
“댓글의 답글에 대한 depth는 최대 몇까지? 의견에 대한 평가는 어떤 식으로? UI/UX는 어떻게?”

토의 및 문서화

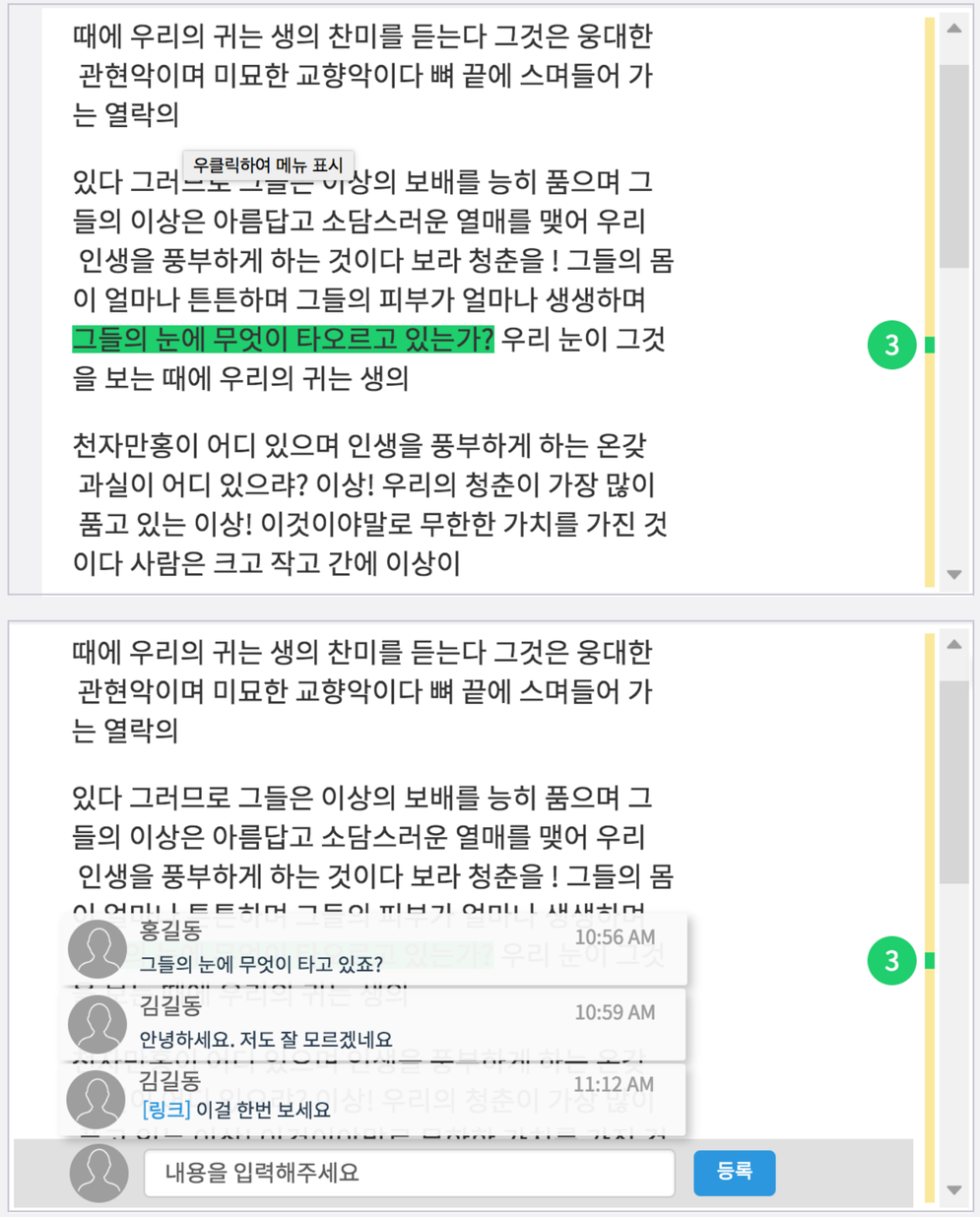
역량강화스
터디

설계
개발할 기능 및 정책

UI



개인화 홈 웹서비스 UI



크롬 익스텐션 UI

설계
개발 추진 전략

전반

웹 전체를 아우르는 의견 공유 시스템

의견 공유를 위한, 서비스의 핵심 기능들(댓글시스템, 개인화 홈 시스템 등)

API 설계 및 구성

DB 설계 및 구성

후반

웹 크롤링 및 머신러닝을 이용한
사이트 추천

your
logo

설계한 내용

크롬 익스텐션 - 컴포넌트와 구현 세부사항 요약

컴포넌트	세부 사항
의견 리스트 보여주기	-로그인을 하지 않아도 의견 리스트 목록을 보여준다. (API) - 나만 보기, 공개 기능 (API) - 의견 리스트 보여줄 때 업다운 표시
의견 삭제하기	- 권한 체크 (API)
의견 달기	- 로그인 했는지 체크 - 작성자 이름, 사진, 마크 다운 텍스트, 작성 시간 (API) - 인풋 창 클릭 시 의견 달기 시작, 어떻게 보여줄 지는 논의 후 결정(TBD) (API)
의견 답글 달기	- 로그인 체크 (API) - 답글인지 여부, 작성자 이름, 사진, 마크 다운 텍스트, 작성 시간 (API) - 답글 depth 1단계

컴포넌트	세부 사항
의견 평가하기, 수정, 삭제	- 로그인 체크 (API) - 권한 체크 (API) - 답글이 있는 댓글 삭제 시 삭제된 댓글입니다 표시 (API) - 업다운 가능하도록 하기 (API) - 수정, 삭제 버튼 - 의견이 수정됐을 때 수정된 글입니다 표시하기
로그인	- 로그인해야 특정 UI 보여줄지는 논의 후 결정(TBD) - 회원가입 클릭 시 - 우리 사이트로 가기 버튼 - 페북, 구글 로그인 (API)
어떤 위치에 의견을 남길 것인지	- 위치 정보, 해당 위치에서 사라지면 페이지에 대한 의견에서 보여주기 (API) - 페이지에 대한 의견 (API)
웹 사이트 평가하기	- 별점 주기, 수정,삭제
알림	-알림을 보여주고 클릭 시 개인화 홈으로 이동

역량강화스터디

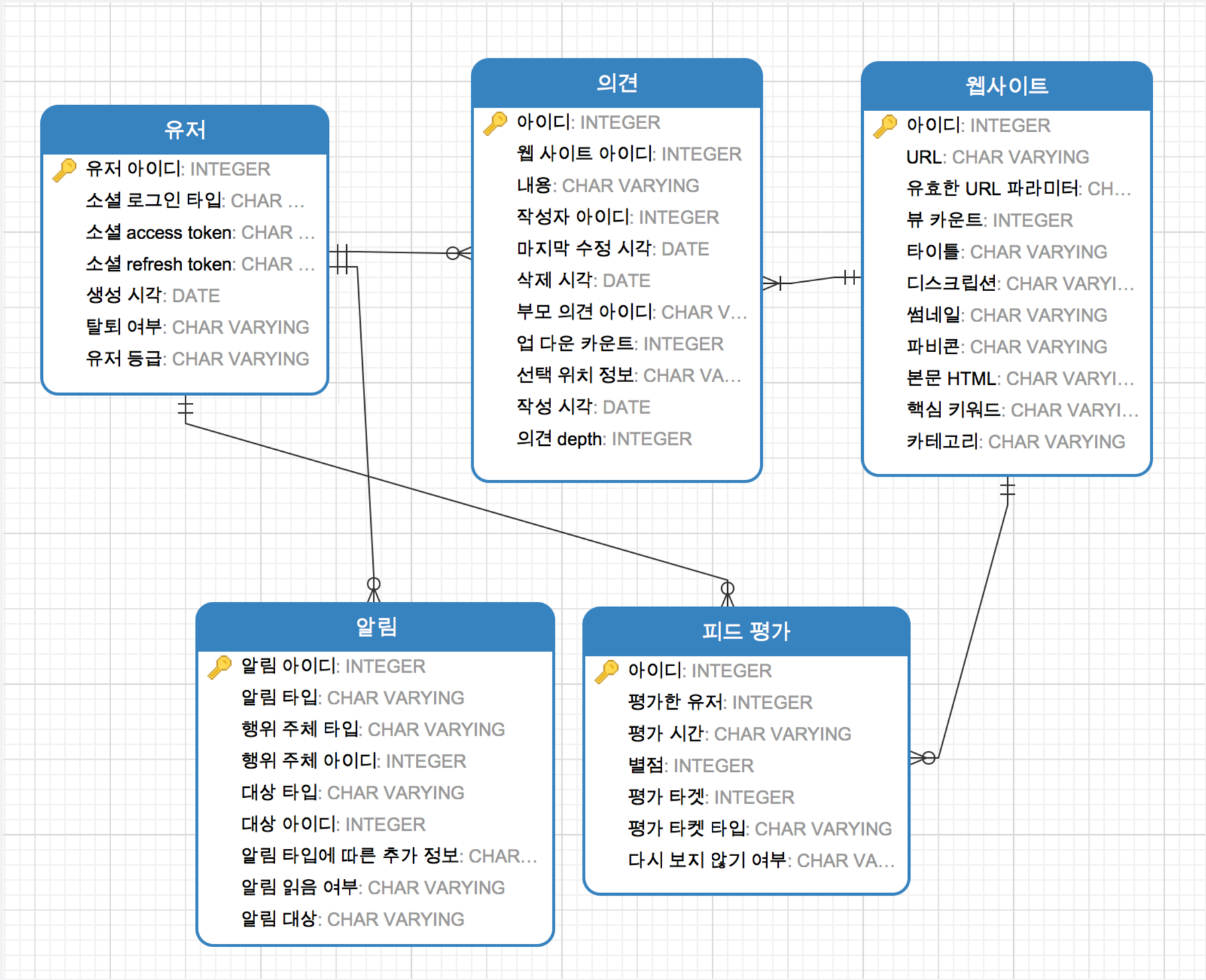
설계한 내용
개인화 홈 - 컴포넌트와 구현 세부사항 요약

컴포넌트	세부 사항
피드 리스트 보여주기	- 피드 의 대상으로 의견이 없는 웹사이트는 포함되지 않는다. - 웹사이트 썸네일, 타이틀, 디스크립션, URL, 평점, 다시 보지 않기 여부 - 피드 의견 클릭 시 해당 페이지에서 의견이 달린 영역으로 이동
웹사이트에 달린 의견 리스트 보여주기	- 피드에 달린 의견 중 업이 높은 것을 보여주기
피드 평가하기	- 별점 UI를 제공하고 왓챠에서 영화를 평가하는 것 처럼 피드를 평가할 수 있도록 하기
피드 다시 보지 않기	- 페이스북에서 제공하는 피드 다시 보지 않기 기능처럼 이 웹사이트 보지 않기, 이 피드 보지 않기, 이런 피드 보지 않기, 이 사용자의 댓글 보지 않기 메뉴 제공하기
회원가입	- 페북, 구글 로그인시 전달된 이름, 프로필 이미지 정보 소셜 아이디를 기반으로 회원가입 처리

컴포넌트	세부 사항
의견에 대해서 크롬 익스텐션과 같은 기능 제공	-로그인을 하지 않아도 의견 리스트 목록을 보여준다. (API) - 나만 보기, 공개 기능 (API) - 의견 리스트 보여줄 때 업다운 표시
로그인	- 로그인해야 특정 UI 보여줄지는 논의 후 결정(TBD) - 회원가입 클릭 시 - 우리 사이트로 가기 버튼 - 페북, 구글 로그인 (API)
알림	- 내가 작성한 의견에 답글이 달렸을 때 A라는 웹사이트에서 내가 남긴 B라는 댓글에 C유저가 D라는 답글을 달았습니다. - 내가 작성한 의견에 다른 유저가 업을 눌렀을 때 A라는 웹사이트에서 내가 남긴 B라는 댓글에 C유저가 업을 눌렀습니다. 알림에서 각각 요소에 하이퍼링크로 이동가능하도록 하기

역량강화스터디

설계한 내용
DB - 모델링



역량강화스
터디

설계한 내용

웹서버 API - 프토토콜 요약

항목	내용	항목	내용
Response 형식	{ "requestUrl": string // Reqeust URL "resultCode": number // Request 결과 코드 "message": string // Request 결과에 대한 메시지 // Request 성공시, 결과 반환 "result": { "total": number // Request 결과에 대한 총 결과 개수 "size": number // 반환된 결과 개수 "offset": number // 반환된 결과의 오프셋 "values": [// 반환된 결과 Raw 데이터 *] } }	**Response 코드**	* 200(성공): 서버가 요청을 제대로 처리함 * 300 : 리다이렉션 * 400(잘못된 요청): 서버가 요청의 구문을 인식하지 못함 * 401(권한 없음) * 403(금지됨): 잘못된 파라미터 사용으로 서버가 요청을 거부함 * 404(찾을 수 없음): 서버가 요청한 페이지를 찾을 수 없음 * 500(내부 서버 오류): 서버에 오류가 발생하여 요청을 수행할 수 없음 * 501(구현되지 않음): 서버에 요청을 수행할 수 있는 기능이 없다. * 503(서비스를 사용할 수 없음)

설계한 내용

웹서버 API - 프토토콜 요약

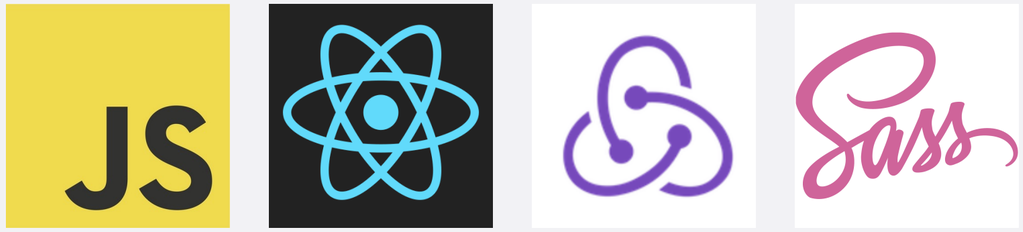
항목	내용
페이지네이션 규칙	offset, limit 파라미터로 페이지네이션을 가능하게 한다. 위 파라미터를 명시하지 않았을 때 기본값은 offset=0, limit=30이다.
인증 규칙	인증이 필요한 http 요청의 header에 Authorization: Bearer {access_token} 을 포함한다. access token은 소셜 로그인의 redirection callback이 유효할 때 refresh token과 함께 발급되며 만료시각은 3일이다. refresh token의 만료시각은 7일 이다.
request format	format : https://{host url}/{api version}/{client type}/{api type}/{target id}?{option parameter} api version은 v1.0과 같이 표시한다. client type은 chrome extension,web이 있다. api type은 comment, feed, notification, auth, website가 있다. target id는 api의 CURD 연산의 타겟이 되는 id이다. option parmeter로는 offset, limit가 있고 get요청의 경우 api에 따라 다르게 정의된다. API는 restful 하게 설계하며 CURD연산에 따라 GET, POST, UPDATE, DELETE 연산을 적절히 사용하도록 한다.
API 문서	https://docs.google.com/spreadsheets/d/14diCHbforEHeSLasfqKBzVfVvPW7ZETxqeo6Z3_eBJQ/edit#gid=0 구글 독스에 각각 API에 대해 http method, request url, request parameter, response를 정의하고 예시를 포함하도록 한다.

설계한 내용

추천 - 수집할 Factor와 사용할 기술

항목	내용
유저 활동 Factor	- 유저의 페이지 별점 입력 - 유저의 피드 별점 입력 - 유저의 피드 클릭 량, 클릭 후 체류 시간 측정 값 - 유저의 브라우저 북마크 - 유저의 웹 사이트 사용 시간대와 자주 방문하는 사이트 - 피드의 재구성 상태 - 유저가 의견에 대해 더 읽기를 클릭했을 때 그 의견의 카테고리와 작성자 정보 - 유저가 의견을 남긴 페이지 - 유저가 의견을 남긴 시간대 - 유저가 입력한 의견의 업, 다운과 평가한 의견 리스트
문서 분석 크롤링	- TF/IDF - 단어, 이미지, 외부 URL, 링크 받은 개수 - 작성자 정보 - 문서의 생서날짜 - 구글 페이지 랭크 정보 - 구글 트렌드에서 평가한 키워드의 중요도
사용할 추천 기술	- collaborative filtering - clustering - regressionanalysis

기술스택
크롬 익스텐션, 개인화 홈



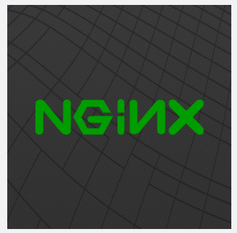
UI와 로직 처리를 위해 **javascript**, **reactjs**, **flux** 아키텍처를 사용하고 css를 구조화하기 위해 **sass**를 사용한다.

자바스크립트를 번들로 만들기 위해 **webpack**을 사용하고 브라우저에서 es6문법을 사용하기 위해 **babel**을 사용한다.

자바스크립트 로직 테스르를 위해 **mocha**를 사용하고 에러를 수집하기 위해 **sentry**를 사용하고 유저 액션을 수집하기 위해 **google analytics**를 사용한다.

역량강화스
터디

기술스택
서버, DB



API 웹 서버를 위해 **nginx**, **nodejs**, **expressjs**를 사용한다.



에러와 서버 모니터링을 위해 **newrelic**을 사용한다. 자체 로그를 수집하고 모니터링하기 위해 **telegraf**, **influxdb**, **grafana**를 사용한다.



데이터를 저장하기 위해 **mongodb**를 사용하고 캐시와 메시지큐로 **redis**를 사용한다. 그리고 이미지와 로그를, static 파일을 저장하기 위해 **amazon s3**를 사용한다.

기술스택
Dev Ops

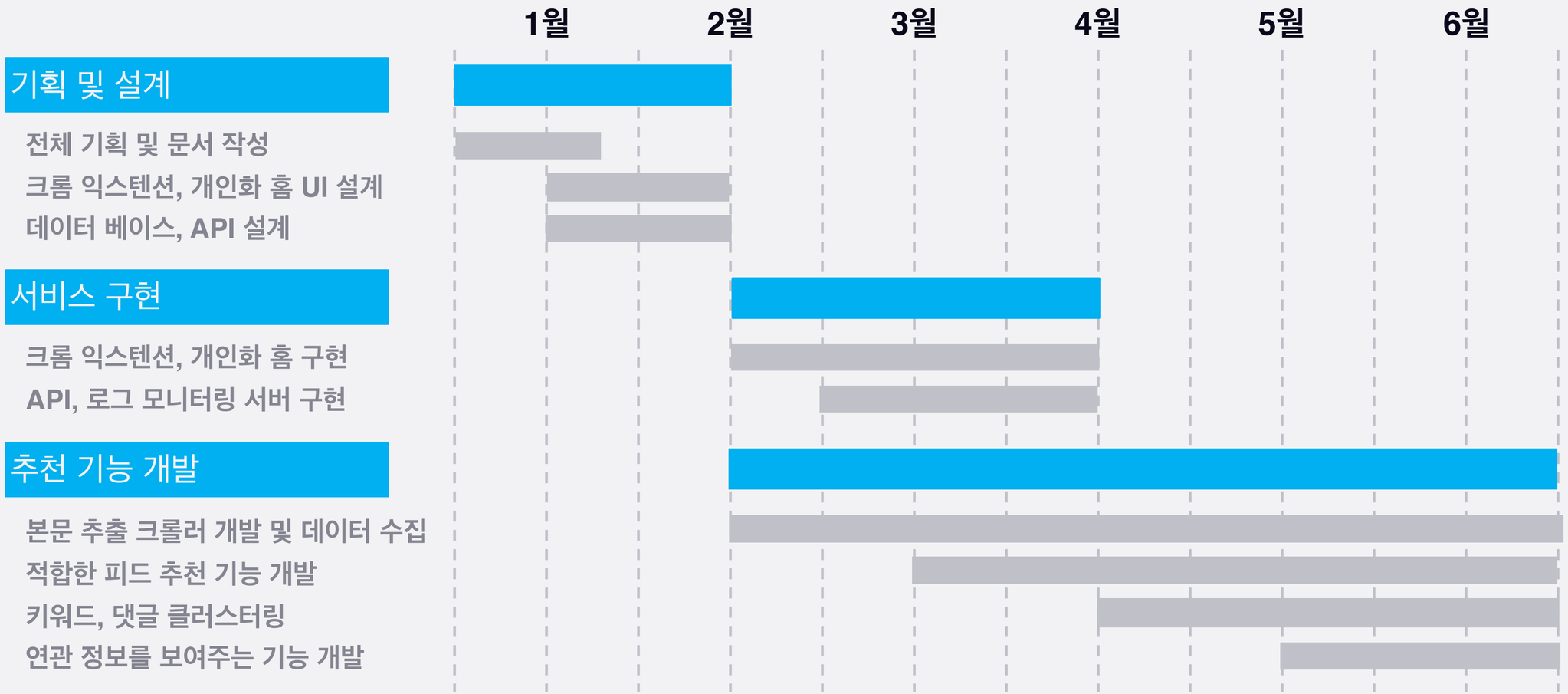
웹 서버와 DB의 배포 자동화를 위해 **ansible**을 사용한다.

프로젝트 형상 관리를 위해 **github**를 사용한다.

테스트 자동화를 위해 **travis-ci**를 사용한다.

추후 계획

개발 일정



역량강화스
터디

Q&A

감사합니다.