

경력기술서

# PROJECT1



## 오늘 어때?

대상 및 기획

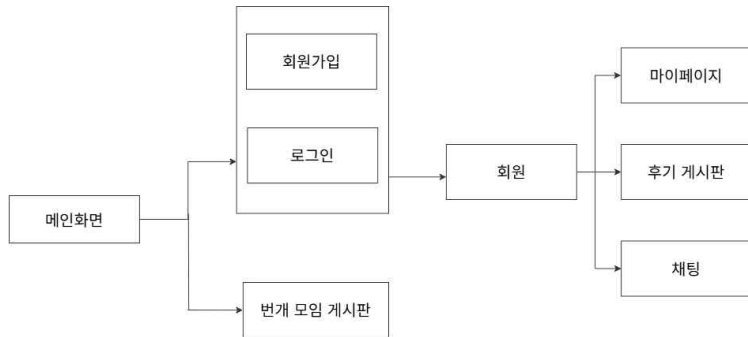
다양한 번개 모임을 즉석으로 모집하고 참여할 수 있는 플랫폼으로 번개모임을 즐겨 찾는 20~40대 직장인 및 대학생들을 대상으로 회원가입을 진행한 회원의 위치와 날씨를 기반으로 취미, 목적 등을 충족 시킬 수 있는 서비스를 기획하였습니다.

구축 및 구현

JSP Model 2 기반으로 웹 서버를 구축하고, 이를 기반으로 동적 웹페이지 생성 및 클라이언트의 요청에 따라 서버 처리와  
JDBC 연동으로 데이터 처리를 구현하였습니다.



# 오늘 어때?



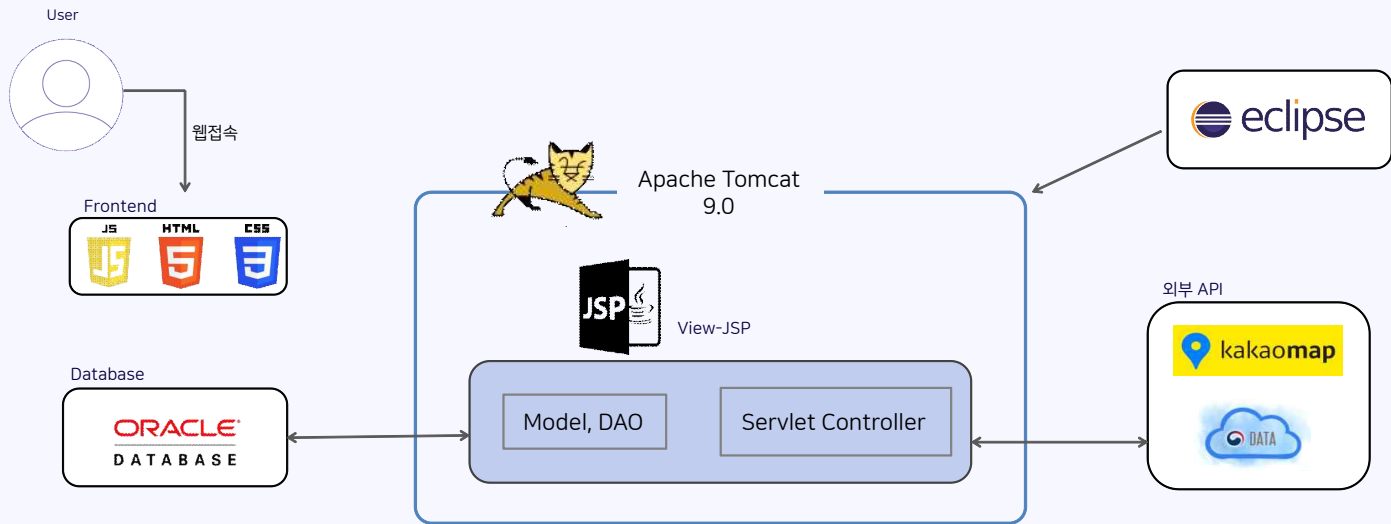
## 오늘 어때? 모임 플랫폼

개발 기간	2025.03.17 ~ 2025.04.07 (5인 팀 프로젝트)
사용 기술	Servlet, JSP Model 2, JSTL, Oracle JDBC, Session HTML5, CSS3, JavaScript Kakao Maps API, 공공 데이터 포털 API
WAS	Apache Tomcat 9.0
OS	Windows 10/11
설계 도구	eXERD, Draw.io
형상 관리	Git, GitHub
Tool	Eclipse, SQL Developer, Discord, Google Drive, Visaul Studio, Adobe PhotoShop, Adobe Illustator



# 오늘 어때?

# 번개모임 플랫폼 - 아키텍처 설계

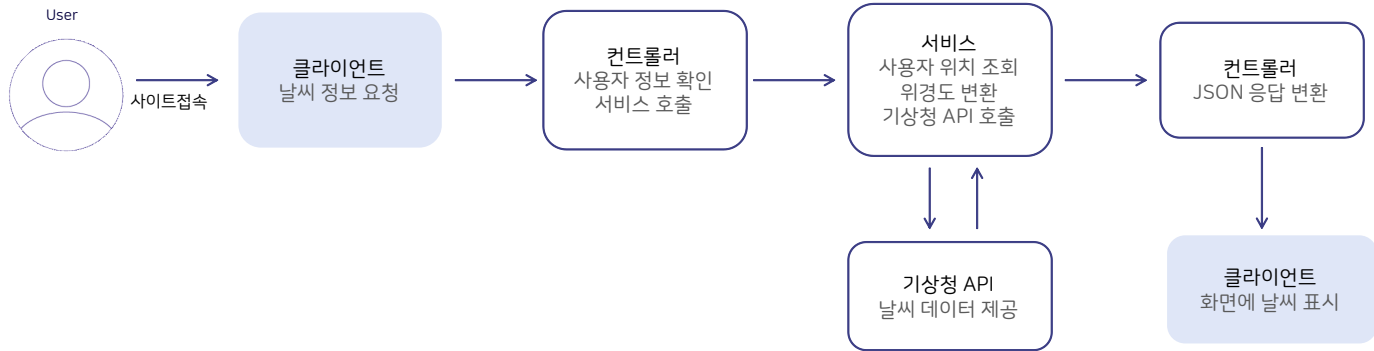




# 오늘 어때?

# 날씨 API 연동 - 순서도

사용기술: Java Servlet, JSP, HTML5, CSS3, JavaScript(AJAX), HttpSession, MVC 패턴, JDBC

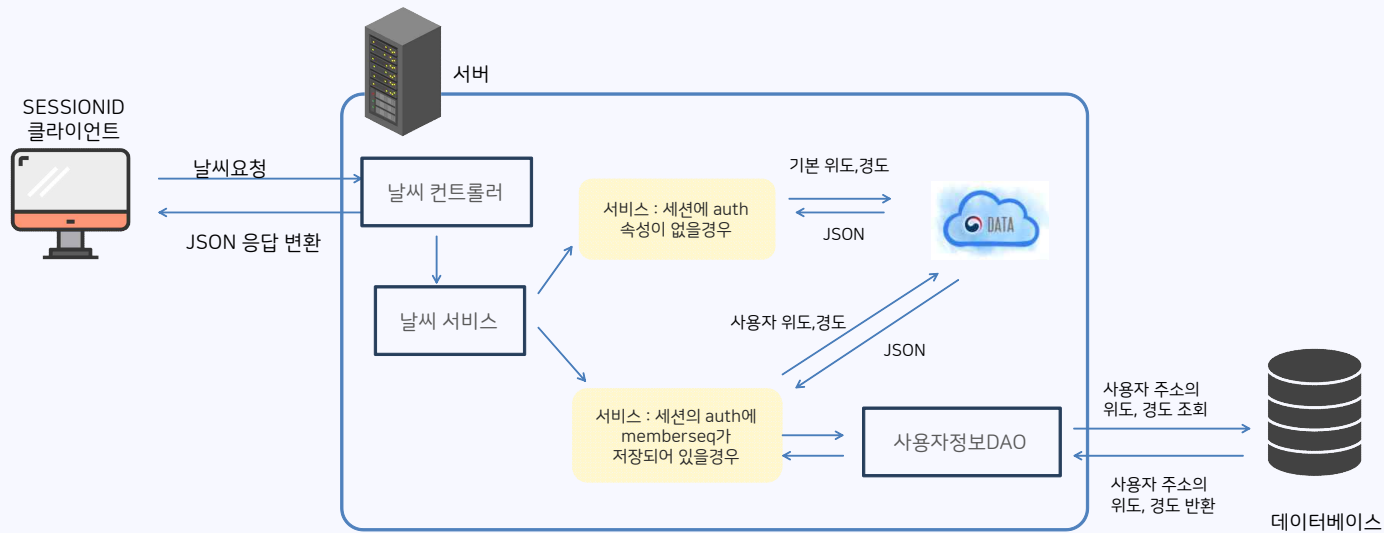




# 오늘 어때?

# 날씨 API 연동 - 아키텍처

사용기술: Java Servlet, JSP, HTML5, CSS3, JavaScript(AJAX), HttpSession, MVC 패턴, JDBC





# 오늘 어때?

# 날씨 API 연동



오늘 어때?

[LOGIN](#) [JOIN](#)

서울특별시 강남구  
온도 : 10.9°C 날씨 : 맑음

카테고리 선택

모임을 검색해 보세요.



맛집 모임



게임 모임



문화 모임



스포츠 모임



자기계발 모임



힐링 모임



1. Ajax를 통해 /weather.do로 서버에 날씨 데이터를 요청하도록 구현했습니다.

2. 현재 시간 기준 한 시간 전 시각을 계산하여, 항상 유효한 기상 정보를 요청하도록 설정했습니다.

3. DB에서 가져온 위도·경도를 기상청 API 요구 형식인 격자 좌표(nx, ny)로 변환하는 로직을 구현했습니다.

4. HttpURLConnection을 통해 기상청 Open API를 호출하고, 응답 코드를 기반으로 성공/에러를 구분하여 안정적인 통신을 처리했습니다.

5. JSON 응답 중 기온(T1H), 강수형태(PHY) 항목만 추출하여 필요한 정보만 가공해 전달했습니다.

6. 로그인 여부에 따라 DB에서 개인 위치를 조회하거나, 기본 좌표를 사용해 날씨 정보를 제공하는 방식으로 개인화된 서비스를 구현했습니다.



## 오늘 어때?

# 날씨 API 연동 - 문제해결

문제점 : 날씨 정보를 제대로 받아오지 못함

서울특별시 강남구

온도 :      날씨

서울특별시 강남구

온도 : 10.9°C 날씨 : 맑음

1. 로그를 통한 문제 파악 : weatherData 객체를 출력해 보니 실제로 빈 JSON만 반환됨을 확인했습니다.

2. 공식 문서·샘플 코드 비교 검토

기상청 Open API 공식 스펙과 예제 코드를 보고 동작 조건을 이해했습니다.

문서에서 “데이터는 발표 시각 이후 일정 시간(예: 10분, 45분 등)이 지나야 조회 가능하다”는 문구를 확인했습니다.

3. 보정 로직 적용 및 실험

“한 시간 전 시간”을 기준으로 요청하도록 보정 로직을 추가해 테스트했습니다.

이 조치로 언제나 준비된 예보 데이터를 안정적으로 받아올 수 있음을 확인했습니다.

4. 단계별 검증 및 문서화

수정된 로직이 전체 서비스에 문제가 없는지 종합 테스트를 실시했습니다.

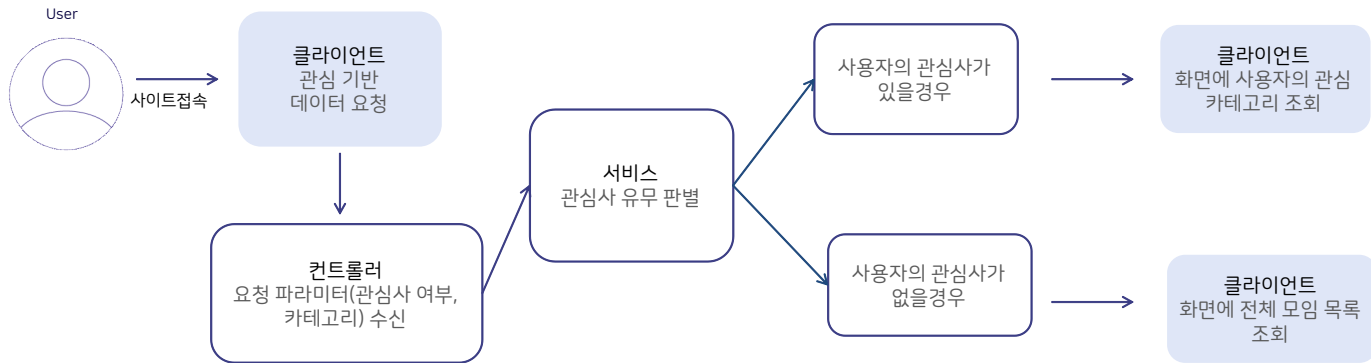
나중에도 의도를 바로 알 수 있도록, 주석과 간단한 설명을 함께 문서화하여 유지보수성을 확보했습니다.



# 오늘 어때?

# 사용자 관심 기반 추천 - 순서도 및 아키텍처

사용기술: Java Servlet, JSP, HTML5, CSS3, JavaScript(AJAX), HttpSession, MVC 패턴, JDBC







# 오늘 어때?

# 사용자 관심 기반 추천

## 만들어진 모임 LIST

[전체보기 >](#)



삼성SDI 코테준비 도  
와주실 분!!!

테카리후베가드아  
2명 모집중



쌍용의전설과 함께  
스파르타 코테스터!

쌍용의전설  
5명 모집중



미장 주식 스터디 팀  
원들 구합니다

일론머스크  
6명 모집중



부동산 스터디 구합  
니다(임장가실 분)

서초진출부  
3명 모집중

## 1. 최고 관심 카테고리 자동 선택

사용자가 카테고리를 지정하지 않거나 첫 방문인 경우, interest\_score에서 점수가 가장 높은 카테고리를 필터로 적용하고, 기록이 없으면 전체 게시글을 기본으로 표시했습니다.

## 2. 검색 히스토리 & 관심도 트래킹

사용자가 검색할 때마다 search\_history와 interest\_score를 동시에 업데이트해 행동 데이터를 실시간으로 집계하고, 이를 추천 정확도 향상에 활용했습니다.

## 3. 서비스/DAO 계층 분리

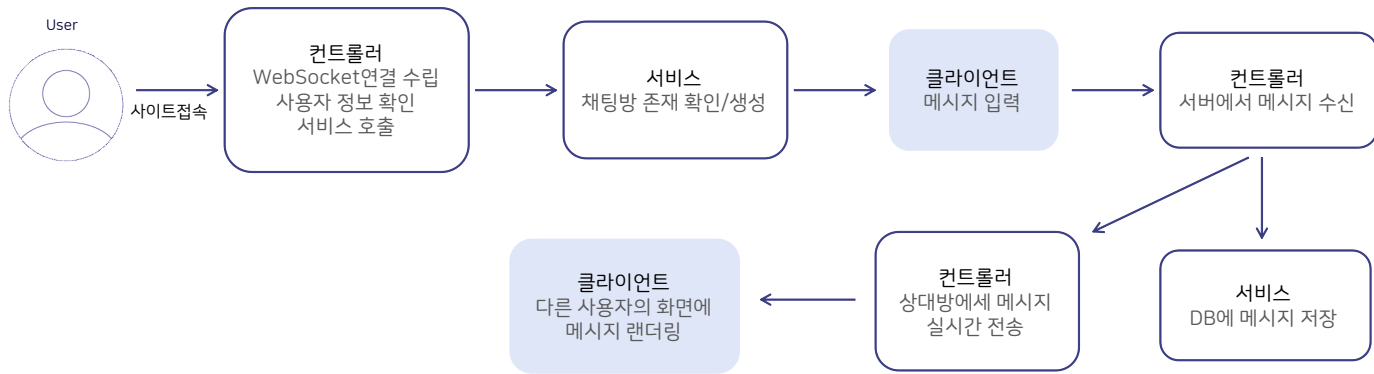
비즈니스 로직은 MainService에, DB 접근은 MainDAO에만 두어 책임을 명확히 분리함으로써 유지보수성과 확장성을 크게 개선했습니다.



# 오늘 어때?

# 웹소켓을 이용한 실시간 채팅 - 순서도

사용기술: Java WebSocket API, Servlet(HttpSession), JSP, Oracle JDBC, WebSocket, AJAX, JSON

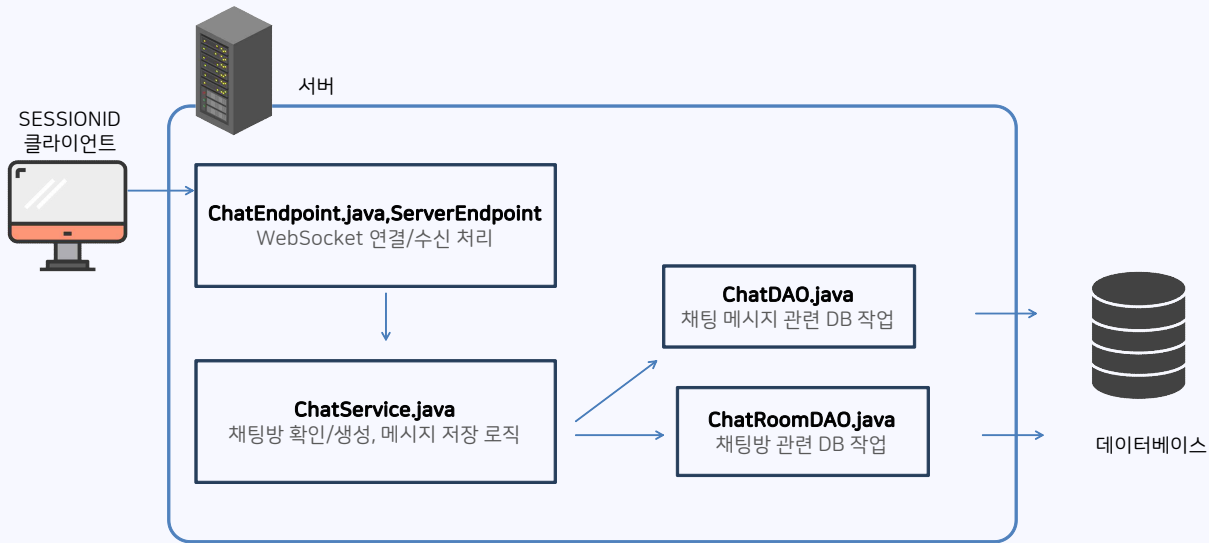




# 오늘 어때?

# 웹소켓을 이용한 실시간 채팅 - 순서도

사용기술: Java WebSocket API, Servlet(HttpSession), JSP, Oracle JDBC, WebSocket, AJAX, JSON





# 오늘 어때?

# 웹소켓을 이용한 실시간 채팅

어때?

닉네임  
채팅방 생성  
돈돈환강03와 채팅하기

나 : 안녕하세요 (오후 10:27:29)  
돈돈환강03 : 안녕하세요 (오후 10:27:34)

메시지를 입력하세요...



오늘 어때?

닉네임  
채팅방 생성  
복복환강00와 채팅하기

복복환강00 : 안녕하세요 (오후 10:27:29)  
나 : 안녕하세요 (오후 10:27:34)

메시지를 입력하세요...

1. 모임에 참여한 회원 간 1:1 실시간 채팅 기능을 구현했습니다.
2. WebSocket을 사용해 페이지 새로고침 없이 메시지를 주고받을 수 있도록 구성했습니다.
3. WebSocket 연결 시 세션에서 사용자 정보를 추출해, 사용자 식별 및 채팅방 생성/조회에 활용했습니다.
4. WebSocket이 끊기거나 세션 정보가 없을 경우를 대비해 예외 상황을 체크하고 조건문을 통해 처리했습니다.
5. 실시간 메시지 전송은 WebSocket, 이전 메시지 조회는 Ajax로 처리하여 사용자 경험을 개선했습니다.



## 오늘 어때?

#느낀점

이번 프로젝트를 통해 MVC 패턴 기반의 구조적 설계가 개발 효율성과 유지보수성에 얼마나 중요한지 깊이 체감할 수 있었습니다. 처음에는 DAO에서 모든 로직을 처리하면서 코드가 복잡하고 중복되는 문제가 많았는데, 이후 MVC 패턴에 맞게 서비스 계층을 도입하자 한눈에 흐름이 파악되고, 로직 수정 시 영향 범위를 명확히 파악할 수 있어 훨씬 안정적인 개발이 가능해졌습니다.

또한 기상청의 RESTful Open API를 연동하는 과정에서 HttpURLConnection을 사용해 데이터를 요청하고, JSONParser로 JSON 응답을 파싱하며, 시간(base\_time)과 좌표(nx, ny)를 보정하는 로직을 구현했습니다. 공식 문서와 외부 자료를 참고해가며, 단순히 데이터를 받아오는 것을 넘어 실제 운영 환경에서 발생할 수 있는 데이터 지연 및 누락 상황까지 고려한 안정적인 설계의 중요성을 배웠습니다.

검색 히스토리 및 관심도 기반 추천 기능을 구현하면서도 사용자의 행동 데이터를 실시간으로 수집하고 반영하는 경험이 무척 흥미로웠고, 데이터 기반 서비스의 가치와 가능성에 대해 큰 흥미를 느꼈습니다.

WebSocket을 활용한 1:1 실시간 채팅 기능 구현을 통해 세션 관리, 메시지 흐름 처리, 동시성 문제 대응 등 실시간 시스템에서 고려해야 할 기술적 요소들을 직접 다뤄보며 많은 것을 배울 수 있었습니다.

마지막으로 협업 과정에서는 Git을 활용한 버전 관리와 더불어 코드 가독성 향상, 주석 작성, 문서화를 통해 팀원 간의 커뮤니케이션 효율을 높이는 방법의 중요성을 몸소 느꼈습니다.