

아동 범죄 예방을 위한  
커뮤니티 및 실시간 위험 정보 제공 플랫폼

# I:ON 아이온

실시간 위험 정보로 지키는 안전



염지원, 박세원, 정지현, 이치우

# 목차

1. 프로젝트 개요

2. 프로젝트 팀 구성 및 역할

3. 프로젝트 수행 절차 및 방법

4. 프로젝트 결과

5. 자체 평가 의견

# 1. 프로젝트 개요

1) 배경 및 개요

2) 기대효과

3) 기술 스택

4) 주요기능



# 1) 프로젝트 배경 및 개요

www.youtube.com > watch :

5년간 아동성범죄 5천100건 '하루 3건꼴' / 연합뉴스TV ..



5년간 아동성범죄 5천100건 '하루 3건꼴' 어린이를  
평균 3건꼴로 일어나는 것으로 나타났습니다.

YouTube · 연합뉴스TV · 2017. 9. 1.



"1만 원 줄 테니까 심부름" 초등생 납치 미수에 불안한 학부모 (자막뉴스) / SBS

조회수 5.1천회 · 4주 전



아동을 대상으로 한 범죄는 매년 꾸준히 발생

-> 자녀를 둔 학부모들의 불안감은 쉽게 사라지지 않고 있음.

위험 지역에 대한 정보나 실종 사건, 성범죄자 거주지 등  
아동 안전과 관련된 데이터는 각각 흩어져 제공되어있음.

지도 기반의 통합적인 정보 제공과 경고 알림 시스템 미흡.

아동 보호자를 대상으로 한  
실시간 위험 정보 제공 플랫폼

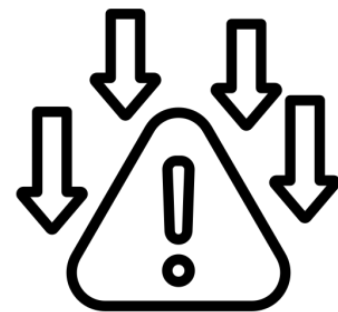
" I\_ON(아이온) " 웹사이트 기획

## 2) 기대효과

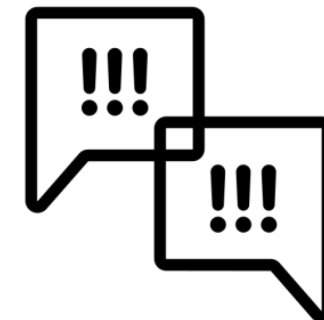
아동 범죄 예방



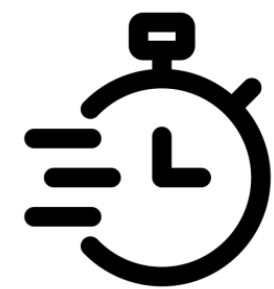
아동 보호자  
불안감 감소



지역 사회  
경각심 향상

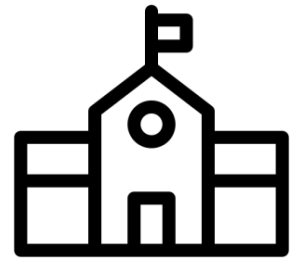


위기 상황  
대응 속도 향상



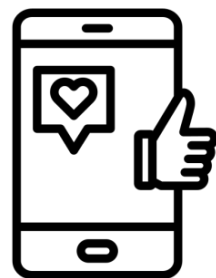
## 2) 기대효과 - 마케팅

### 학교·유치원 연계 홍보



가정통신문, 학부모회 안내문에  
QR 코드 삽입

### SNS 카드뉴스



범죄 예방 팁 + 플랫폼 기능 요약  
(인스타, 카카오톡 채널)

### 신규 가입 프로모션



가입시 아동 범죄 구간 지도 pdf 증정  
등 신규 가입 시 혜택

### 지역 커뮤니티 활용



“아이 안전을 지키는  
무료 앱/웹”으로 접근

## 2) 기대효과 - 영업이익

### 프리미엄 멤버십

무료 기본 기능

: 실시간 위험 알림, 게시판

유료 기능

: 사용자 지역 기반 안전 범위  
안내, AI 범죄 예측 제공

### 안전용품 제휴 판매

아동 위치 추적기,  
안전 호루라기, 스마트 워치  
등 연계 판매

→ 판매 수수료 수익

### 지역 맞춤형 광고

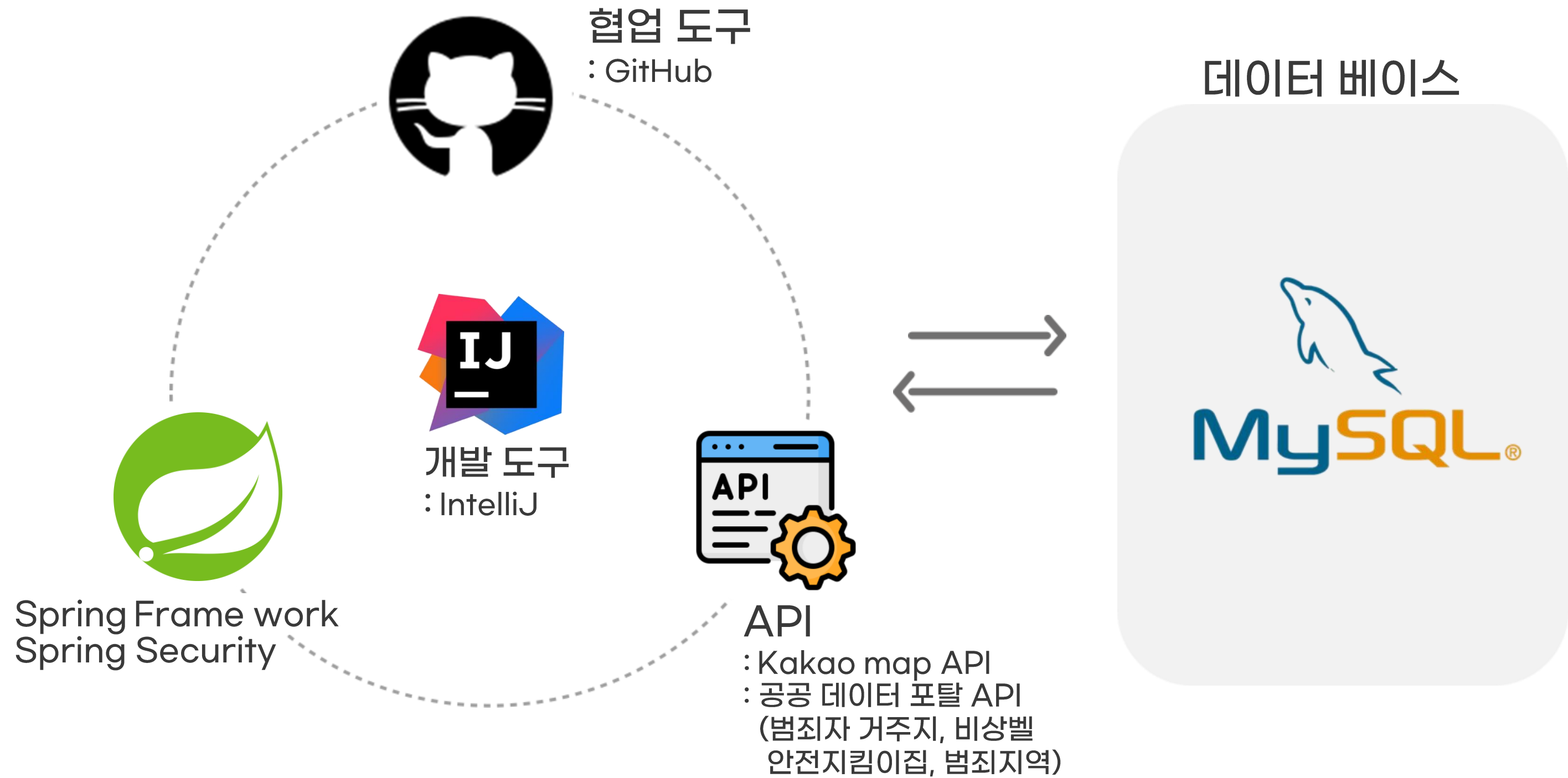
학부모 대상 교육·보험·안전  
제품 광고 노출

### 지자체 공식 플랫폼 계약

지자체 공식 안전 알림망으로  
계약

→ 연간 유지보수 비용 수익

### 3) 기술 스택





## 4) 주요 기능

01

관리자 회원 관리(재학생 인증, 신고, 회원 정지)

02

커뮤니티(자유, 위탁, 실종 신고), 제보 및 신고, 정보 공유 게시판

03

실시간 쪽지 및 알림 팝오버

04

지역 사건 발생 알림 및 댓글 알림

05

범죄 주의 구간, 성범죄자 거주지, 비상벨, 안전지킴이 집 지도



## 2. 프로젝트 팀 구성 및 역할

### 염지원(팀장)

- 전체 일정 및 작업 분배 관리
- 마이페이지(지역 설정, 프로필 이미지, 활동 내역 표시 구현)
- 신뢰도 점수 및 배지 시스템 구현
- 제보 및 신고 게시판 구현 및 관리자 승인 플로우 구축

### 박세원

- 아동 범죄 예방 게시판 구현
- 마이페이지 내 소식  
(지역 기반 사건 알림, 댓글 알림 구현)
- 알림 팝오버(사건, 댓글) 구현
- 사용자 간 프로필 열람 기능 구현
- 발표 자료 준비 및 발표

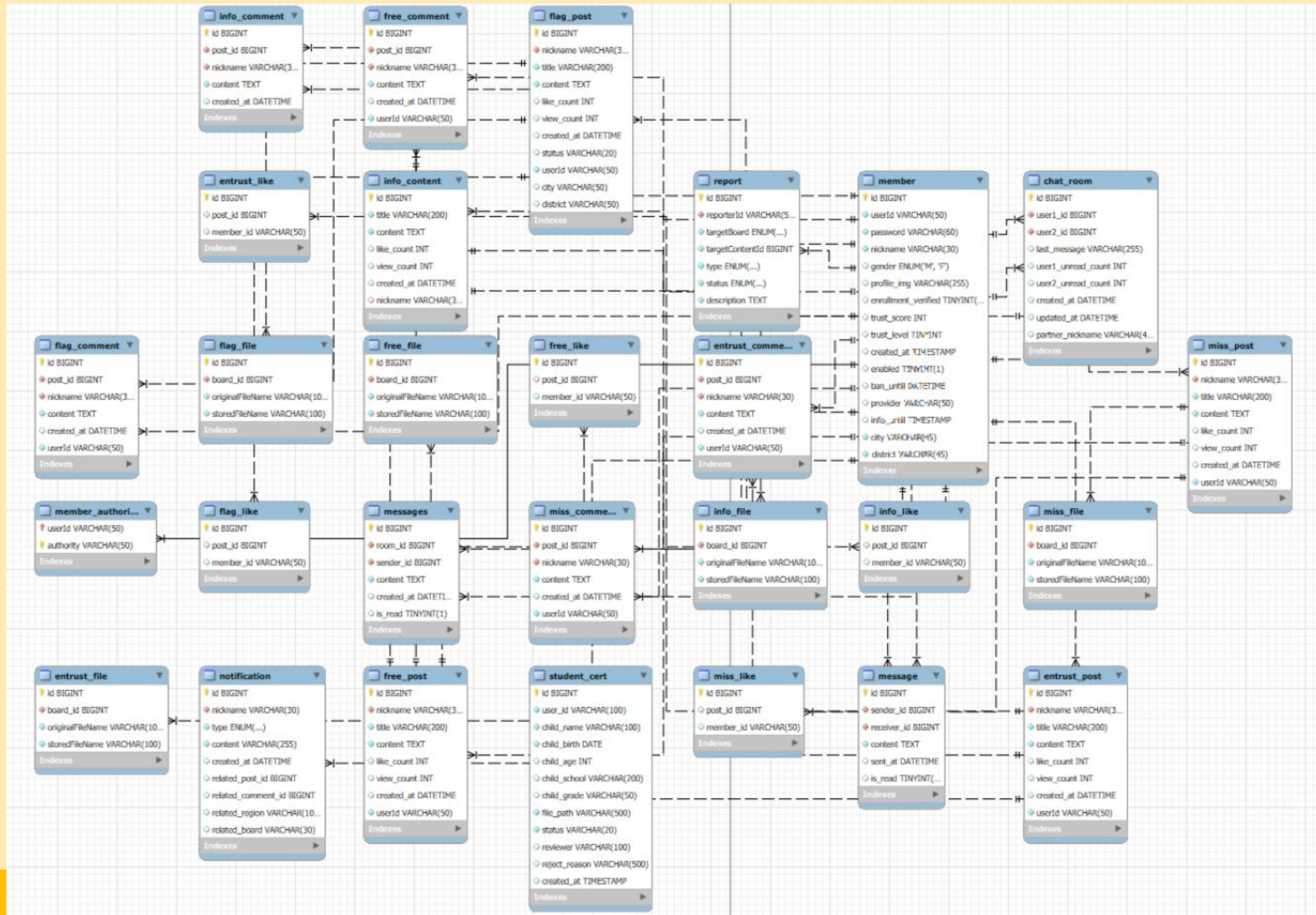
### 이치우

- 회원가입/로그인 시스템 구현 및
- 네이버 로그인 연동
- 인증 여부에 따라 게시판 기능 제한
- 웹 소켓을 활용한 실시간 채팅 기능 구현
- 기타 보안 기능 적용
- Spring Security 커스텀 설정

### 정지현

- 커뮤니티(자유, 위탁, 실종) 게시판 구현
- 게시글 신고 기능 구현
- 성범죄자 거주지, 지킴이집, 비상벨, 범죄발생 구역 API 연동 처리
- 범죄 지역 시각화 및 위험 경고 표시
- 회원 정지 구현

# 3. ER 다이어그램



## 4. 프로젝트 결과

- 1) 시연 영상
- 2) API
- 3) 신뢰도 차트
- 4) 알림 팝오버
- 5) Security
- 6) 실시간 채팅 기능





# 1) 시연 영상

← → ↻ 📄 localhost:8080/login ☆ 📦 | 📶 시크릿 모드 ⋮

아동 범죄 예방을 위한 커뮤니티 및 실시간 위험 정보 제공 플랫폼

**I:ON 아이온**  
실시간 위험 정보로 지키는 안전

### 아이온 통합 로그인

안전한 사용을 위해 로그인해 주세요

아이디

아이디

비밀번호

비밀번호

☐ 로그인 유지

로그인

또는

**N** 네이버로 로그인하기

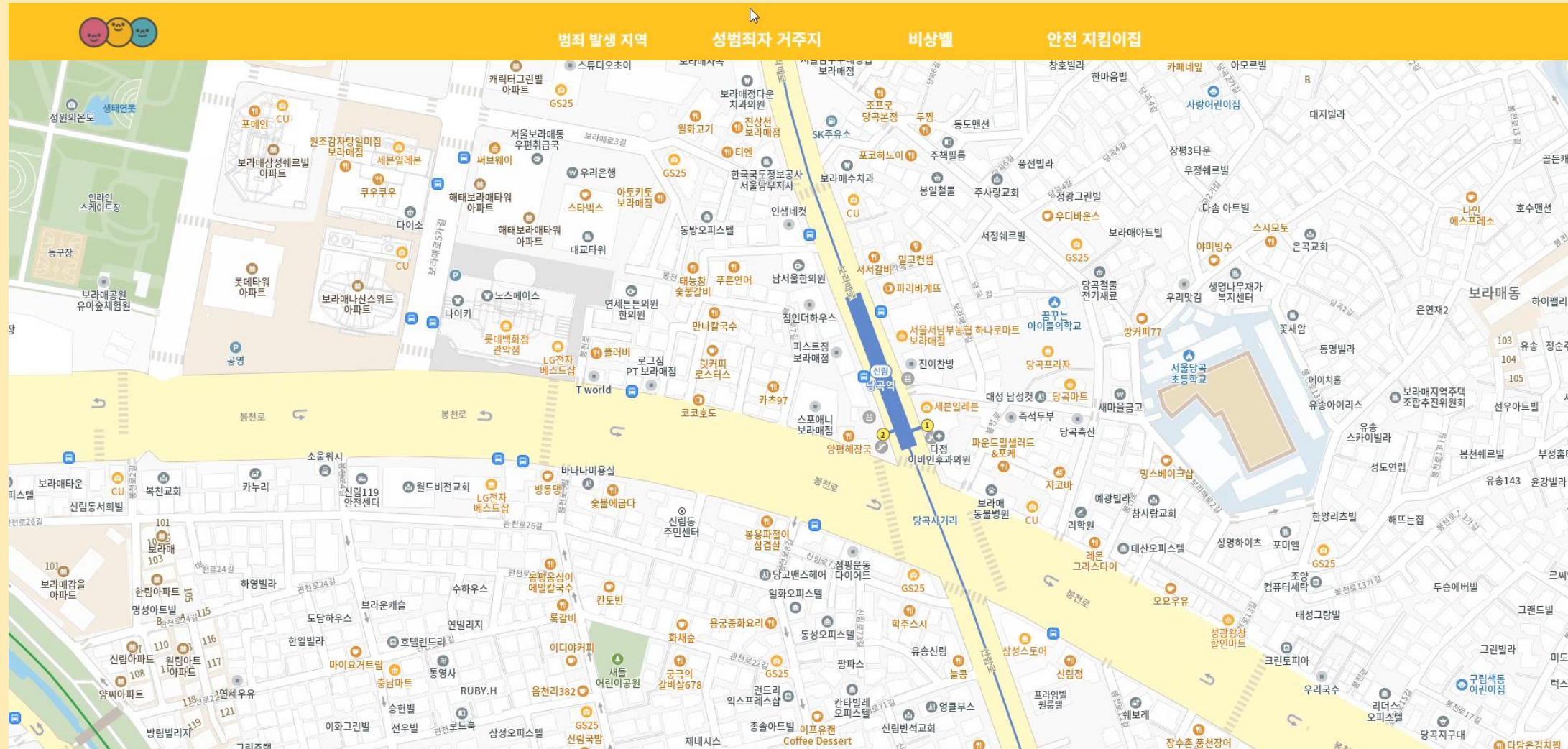
회원가입하기

네이버 회원가입



kakao map

## 2) API



- 성범죄자 거주지, 비상벨, 안전 지킴이집 (카카오맵)
- 범죄 발생 지역 (api에서 제공한 지도로 구현): 카카오 맵에서 wms 방식 지원x

```
let emergencyMarkers = [];  
  
window.clearEmergencyMarkers = function () {  
  if (window.clusterer) {  
    window.clusterer.removeMarkers(emergencyMarkers);  
  }  
  emergencyMarkers.forEach(mk => mk.setMap(null));  
  emergencyMarkers = [];  
};  
  
window.loadEmergencyMarkersByBounds = function () {  
  if (!window.map) return;  
  
  const bounds = window.map.getBounds();  
  const sw = bounds.getSouthWest();  
  const ne = bounds.getNorthEast();  
  
  const url = ` /map/emergencybell?swLat=${sw.getLat()}&swLng=${sw.getLng()}&neLat=${ne.getLat()}&neLng=${ne.getLng()} `;  
  
  fetch(url)  
    .then(res => res.json())  
    .then(data => {  
      window.clearEmergencyMarkers();  
  
      const newMarkers = [];  
  
      data.forEach(marker => {  
        const { latitude, longitude } = marker;  
        if (!latitude || !longitude || latitude === 0 || longitude === 0) return;  
  
        const position = new kakao.maps.LatLng(latitude, longitude);  
  
        const emergencyMarkerImage = new kakao.maps.MarkerImage(  
          '/resources/img/emergencybell-marker.png',  
          new kakao.maps.Size(32, 32)  
        );  
  
        const mk = new kakao.maps.Marker({  
          position,  
          image: emergencyMarkerImage  
        });  
  
        attachPopup(marker, mk, "emergency");  
        emergencyMarkers.push(mk);  
        newMarkers.push(mk);  
      });  
  
      if (window.clusterer && newMarkers.length > 0) {  
        window.clusterer.addMarkers(newMarkers);  
      }  
    })  
    .catch(err => console.error("❗ 비상벨 마커 로딩 실패:", err));  
};
```





### 3) 신뢰도 차트

```
@Repository 2개 사용 위치 ㄴjiwon
@RequiredArgsConstructor
public class TrustScoreRepository {
    private final SqlSessionTemplate sql;

    public int countAllCommentsByNickname(String nickname) { 1개 사용 위치 ㄴjiwon
        int free = sql.selectOne( statement: "trust.countFreeComments", nickname);
        int flag = sql.selectOne( statement: "trust.countFlagComments", nickname);
        int miss = sql.selectOne( statement: "trust.countMissComments", nickname);
        int entrust = sql.selectOne( statement: "trust.countEntrustComments", nickname);
        return free + flag + miss + entrust;
    }

    public int countAllReportsByNickname(String nickname) { 1개 사용 위치 ㄴjiwon
        return sql.selectOne( statement: "trust.countAllReportsByNickname", nickname);
    }

    public int countAllEntrustsByNickname(String nickname) { 1개 사용 위치 ㄴjiwon
        return sql.selectOne( statement: "trust.countAllEntrustsByNickname", nickname);
    }

    public void updateMemberTrust(String nickname, int total) { 1개 사용 위치 ㄴjiwon
        var param = new java.util.HashMap<String, Object>();
        param.put("nickname", nickname);
        param.put("total", total);

        sql.update( statement: "trust.updateMemberTrustScoreOnly", param);
    }
}
```

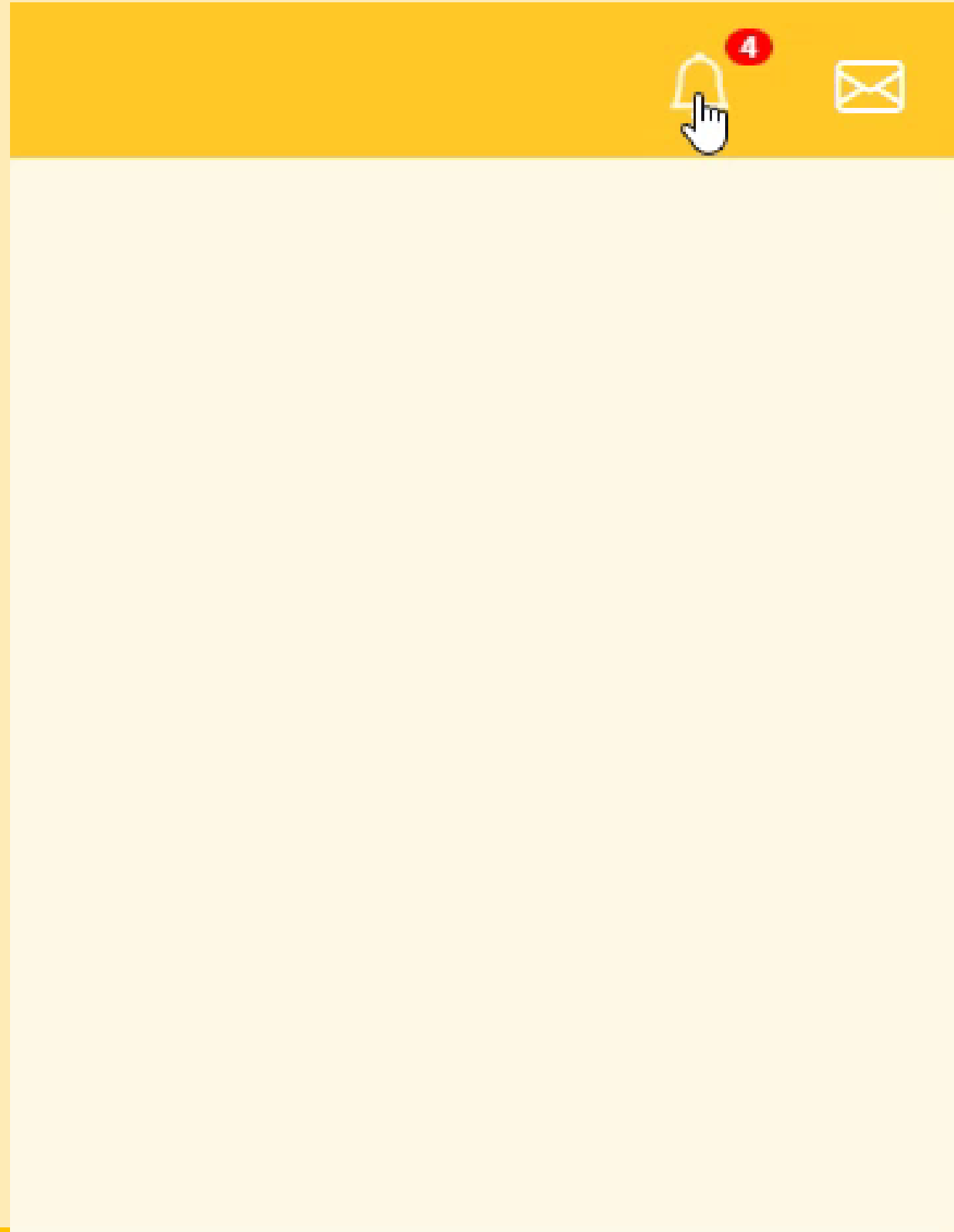


```
const reportCount = Number(`${trustScore.reportCount}`);
const entrustCount = Number(`${trustScore.entrustCount}`);
const commentCount = Number(`${trustScore.commentCount}`);
const ctx = document.getElementById('trustDonut').getContext('2d');
new Chart(ctx, {
    type: 'doughnut',
    data: {
        labels: ['제보', '위탁', '댓글'],
        datasets: [{
            data: [reportCount, entrustCount, commentCount],
            backgroundColor: ['#4bc0c0', '#f6a623', '#63a4fa'],
            borderWidth: 0
        }]
    },
    options: {
        cutout: '65%',
        plugins: { legend: {display: false}, tooltip: {callbacks: {label: (c) => c.label + ': ' + c.raw + '개'}} }
    }
});

const totalScore = Number(`${trustScore.totalScore}`);
const grade = `${fn.trim(trustScore.grade)}`;
const gaugeBar = document.getElementById('trustGaugeBar');
const gaugeText = document.getElementById('trustGaugeText');
const maxScore = 30;
const percent = Math.min((totalScore / maxScore) * 100, 100);
setTimeout(() => {gaugeBar.style.width = percent + '%';}, 300);

let text = '';
if (grade === '최고 안전 수호자') {
    text = '최고 등급 달성! 🏆';
} else if (grade === '안심 지킴이') {
    text = '최고 안전 수호자까지 <b> + Math.max(0, 30 - totalScore) + '점 남았어요!';
} else {
    text = '안심 지킴이까지 <b> + Math.max(0, 10 - totalScore) + '점 남았어요!';
}
gaugeText.innerHTML = text;
</script>
```

- 배지 변경
- 제보/ 위탁/ 댓글 작성 시 1점 획득



```
<script>
document.addEventListener("DOMContentLoaded", async function () {
  var btn = document.getElementById("alertBtn");
  if (!btn) return;
  var base = `${pageContext.request.contextPath}` || "";

  try {
    var res = await fetch(base + "/notify/list", { credentials: "same-origin" });
    if (!res.ok) throw new Error("HTTP " + res.status);
    var items = await res.json();

    var unreadCount = items.filter(n => !n.isRead).length;
    var badge = document.getElementById("notify-unread-count");
    if (badge) {
      if (unreadCount > 0) { badge.textContent = unreadCount; badge.style.display = "inline-block"; }
      else { badge.style.display = "none"; }
    }

    items.sort(function (a, b) { return (b.created_at || 0) - (a.created_at || 0); });

    var html = '<div style="max-height:220px; overflow-y:auto;"><ul class="list-unstyled mb-0">';
    if (items.length) {
      for (var i = 0; i < items.length; i++) {
        var n = items[i];
        var text = (n && n.content) ? n.content : '';
        var board = (n && n.related_board) ? n.related_board : '';
        var pid = (n && n.related_post_id) ? n.related_post_id : '';
        var link = base + "/" + board + "/" + pid;

        html += '<li style="padding:6px 0; border-bottom:1px solid #f0f0f0;">'
          + '<a href="' + link + '" style="text-decoration:none; color:inherit;">' + text + '</a>'
          + '<button type="button" class="notify-delete" data-id="' + n.id + '" '
          + 'style="margin-left:8px;background:none;border:none;cursor:pointer;">✕</button>'
          + '</li>';
      }
    } else {
      html += '<li style="padding:10px;">알림이 없습니다 😊</li>';
    }
    html += '</ul></div>';
  }
});
```

```
const pop = new bootstrap.Popover(btn, {
  html: true, container: 'body', placement: 'bottom', trigger: 'click',
  title: '알림', content: html, sanitize: false, customClass: 'notify-panel', offset: [0, 15]
});

} catch (e) {
  console.error("notify popover error:", e);
  var fallback = '<div style="max-height:220px; overflow-y:auto;">'
    + '<ul class="list-unstyled mb-0"><li>알림을 불러오지 못했습니다</li></ul>'
    + '</div>';
  new bootstrap.Popover(btn, {
    html: true, container: 'body', placement: 'bottom', trigger: 'click',
    title: '알림', content: fallback
  });
}

});

document.addEventListener('click', function(e){
  if (e.target && e.target.classList.contains('notify-delete')) {
    const id = e.target.getAttribute('data-id');
    deleteNotify(id);
    e.target.closest('li')?.remove();
  }
});

function deleteNotify(id, buttonElement) {
  if (!confirm("이 알림을 삭제할까요?")) return;
  const base = `${pageContext.request.contextPath}` || "";
  fetch(base + "/notify/delete/" + id, {
    method: "DELETE", credentials: "same-origin"
  })
  .then(res => {
    if (!res.ok) throw new Error("삭제 실패: " + res.status);
    return res.text();
  })
  .then(() => { window.location.reload(); })
  .catch(err => {
    console.error(err);
    alert("삭제 중 오류가 발생했습니다.");
  });
});
</script>
```

-팝오버 생성 , 알림 내역, 알림 배지





## 5) Security

```
<beans xmlns="http://www.springframework.org/schema/beans"
  <security:http use-expressions="true">
    <security:intercept-url pattern="/accessDenied" access="permitAll"/>
    <security:intercept-url pattern="/" access="isAuthenticated()"/>
    <security:intercept-url pattern="/free/**" access="isAuthenticated()"/>
    <security:intercept-url pattern="/map/**" access="isAuthenticated()"/>
    <security:intercept-url pattern="/info/**" access="isAuthenticated()"/>
    <security:intercept-url pattern="/withdraw" access="isAuthenticated()"/>
    <security:intercept-url pattern="/flag/**" access="hasAnyRole('ROLE_AUTH_MEMBER', 'ROLE_ADMIN')"/>
    <security:intercept-url pattern="/entrust/**" access="hasAnyRole('ROLE_AUTH_MEMBER', 'ROLE_ADMIN')"/>
    <security:intercept-url pattern="/miss/**" access="hasAnyRole('ROLE_AUTH_MEMBER', 'ROLE_ADMIN')"/>
    <security:intercept-url pattern="/admin/**" access="hasRole('ROLE_ADMIN')"/>
    <security:intercept-url pattern="/register" access="isAnonymous()"/>
    <security:intercept-url pattern="/login" access="permitAll"/>

    <security:form-login login-page="/login" default-target-url="/"
      authentication-failure-handler-ref="customLoginFailHandler"/>

    <security:logout logout-url="/logout" logout-success-url="/" invalidate-session="true"
      delete-cookies="JSESSIONID, remember-me"/>

    <security:remember-me token-validity-seconds="3600"/>

    <security:access-denied-handler ref="customAccessDeniedHandler"/>

    <security:session-management invalid-session-url="/login">
      <security:concurrency-control max-sessions="1" expired-url="/login" session-registry-ref="sessionRegistry"
        error-if-maximum-exceeded="false"/>
    </security:session-management>

    <security:csrf disabled="true"/>

    <security:headers>
      <security:cache-control/>
    </security:headers>

  </security:http>

  <security:websocket-message-broker>
    <security:intercept-message pattern="/chat/**" access="permitAll"/>
    <security:intercept-message pattern="/sub/chat/room/**" access="isAuthenticated()"/>
    <security:intercept-message pattern="/pub/chat/send" access="isAuthenticated()"/>
  </security:websocket-message-broker>

  <security:authentication-manager>
    <security:authentication-provider user-service-ref="customUserDetailsService">
      <security:password-encoder ref="passwordEncoder"/>
    </security:authentication-provider>
  </security:authentication-manager>
</security:authentication-manager>
```

아동 범죄 예방을 위한 커뮤니티 및 실시간 위험 정보 제공 플랫폼

I:ON 아이온  
실시간 위험 정보로 지키는 안전

### 아이온 통합 로그인

안전한 사용을 위해 로그인해 주세요

아이디

아이디

이 입력칸을 작성하세요

비밀번호

비밀번호

☐ 로그인 유지

로그인

또는

N 네이버로 로그인하기

회원가입하기

```
@Service 1개 사용 위치 조옮
@RequiredArgsConstructor
public class CustomUserDetailsService implements UserDetailsService {

    private final MemberRepository memberRepository;

    @Override 0개씩 사용위치 조옮
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        MemberDTO member = memberRepository.findByIdWithAuthorities(username);
        if (member == null) {
            throw new UsernameNotFoundException("사용자를 찾을 수 없습니다: " + username);
        }

        if (member.getBanUntil() != null) {
            if (LocalDateTime.now().isAfter(member.getBanUntil())) {
                member.setEnabled(true);
                memberRepository.unban(member);
            } else {
                DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy년 MM월 dd일 HH시 mm분");
                throw new DisabledException(member.getBanUntil().format(formatter) + "까지 비활성화된 계장입니다.");
            }
        }

        return new CustomUserDetails(member);
    }
}
```

```
public class CustomLoginFailHandler extends SimpleUrlAuthenticationFailureHandler { 1개 사용 위치 조옮

    @Override 1개 사용 위치 조옮
    public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse response, AuthenticationException exception) throws IOException, ServletException {
        String loginError;

        if (exception instanceof LockedException || (exception.getCause() != null && exception.getCause() instanceof LockedException)) {
            loginError = "잠긴 계장입니다.";
        } else if (exception instanceof DisabledException || (exception.getCause() != null && exception.getCause() instanceof DisabledException)) {
            loginError = exception.getMessage();
        } else if (exception instanceof AccountExpiredException || (exception.getCause() != null && exception.getCause() instanceof AccountExpiredException)) {
            loginError = "만료됨 계장입니다.";
        } else if (exception instanceof CredentialsExpiredException || (exception.getCause() != null && exception.getCause() instanceof CredentialsExpiredException)) {
            loginError = "비밀번호가 만료되었습니다.";
        } else if (exception instanceof BadCredentialsException || (exception.getCause() != null && exception.getCause() instanceof BadCredentialsException)) {
            loginError = "아이디 또는 비밀번호가 틀립니다.";
        } else if (exception instanceof UsernameNotFoundException || (exception.getCause() != null && exception.getCause() instanceof UsernameNotFoundException)) {
            loginError = "존재하지 않는 사용자입니다.";
        } else {
            loginError = "알 수 없는 이유로 로그인에 실패했습니다.";
        }

        request.getSession().setAttribute("loginError", loginError);
        setDefaultFailureUrl("/login");
        super.onAuthenticationFailure(request, response, exception);
    }
}
```

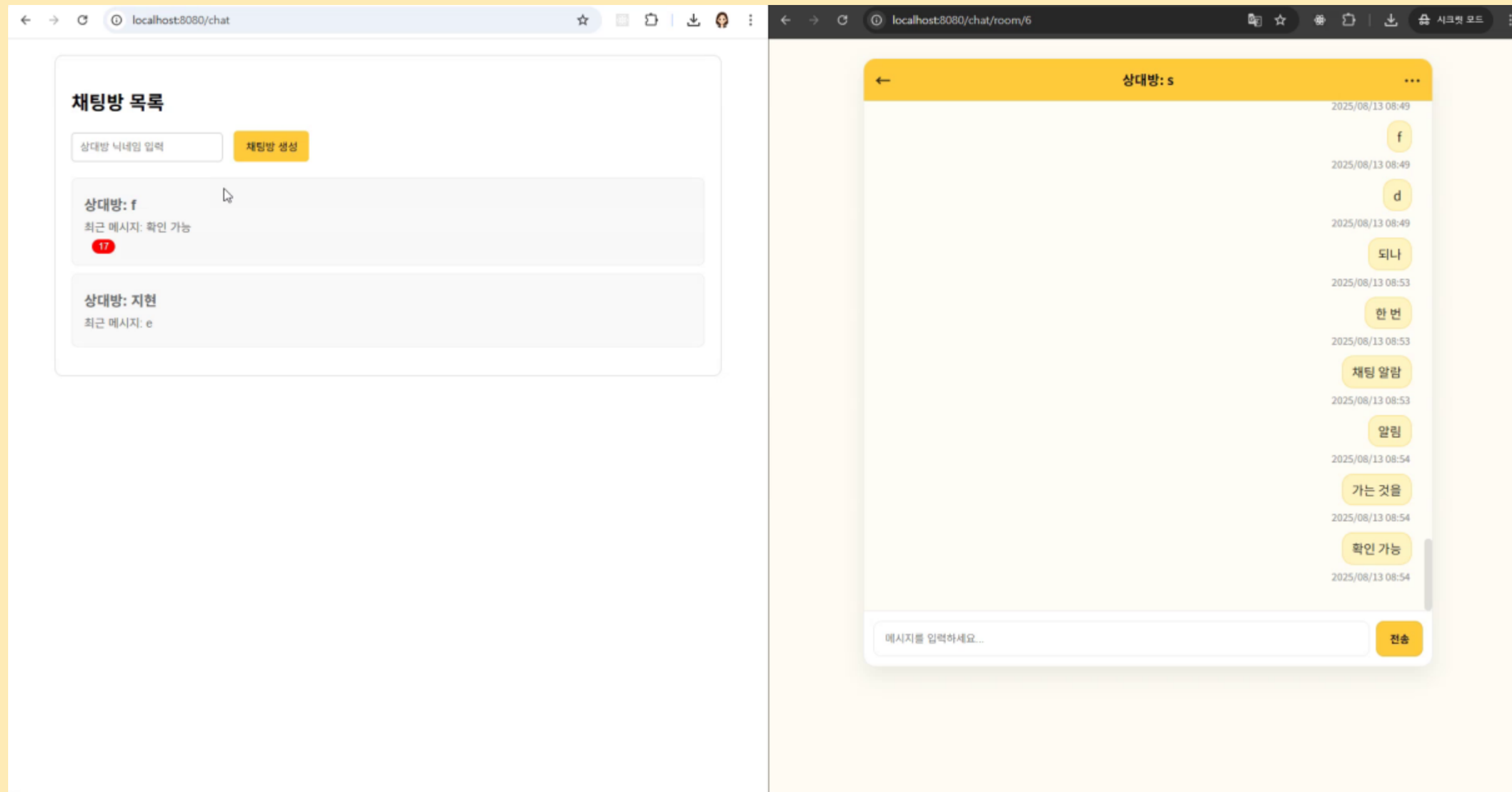
접근 제한 (Ban) 회원 로그인 시,

커스텀한 예외 발생

=> 로그인 제한



## 6) 실시간 채팅



```
@MessageMapping("/chat/send") @개회 사용위치 @LCW
public void sendMessage(MessageDTO message) {
    message.setIsRead(false);
    chatRoomService.updateLastMessage(message.getRoomId(), message.getContent());
    MessageDTO sentMessage = messageService.saveMessage(message);

    LocalDateTime createdAt = sentMessage.getCreatedAt();
    if (createdAt != null) {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm");
        String formattedDate = createdAt.format(formatter);
        sentMessage.setFormattedCreatedAt(formattedDate);
    }

    ChatRoomDTO chatRoom = chatRoomService.findChatRoomById(message.getRoomId());
    if (chatRoom == null) {
        return;
    }

    Long partnerId = chatRoom.getUser1Id().equals(message.getSenderId()) ? chatRoom.getUser2Id() : chatRoom.getUser1Id();

    boolean isPartnerInRoom = activeUsers.containsKey(partnerId)
        && activeUsers.get(partnerId).equals(message.getRoomId());

    if (!isPartnerInRoom) {
        chatRoomService.incrementUnreadCount(message.getRoomId(), message.getSenderId());
    }

    messagingTemplate.convertAndSend(destination: "/sub/chat/room/" + message.getRoomId(), sentMessage);

    messagingTemplate.convertAndSend(destination: "/sub/chat/user/" + partnerId, sentMessage);
    messagingTemplate.convertAndSend(destination: "/sub/chat/user/" + message.getSenderId(), sentMessage);

    String partnerPrincipalName = memberService.findById(partnerId).getNickname();
    String senderPrincipalName = memberService.findById(message.getSenderId()).getNickname();
    messagingTemplate.convertAndSendToUser(partnerPrincipalName, destination: "/sub/chat/user/" + partnerId, sentMessage);
    messagingTemplate.convertAndSendToUser(senderPrincipalName, destination: "/sub/chat/user/" + message.getSenderId(), sentMessage);
}
```

웹 소켓 사용하여 실시간 채팅 구현

## 5. 프로젝트 확장 방안

### 앱(APP) 출시

웹 기반에서 모바일 앱 (Android/iOS) 으로 확장 푸시알림 강화

### AI 범죄 위험 예측

과거 범죄 데이터,  
위치 데이터 분석

"이 시간대, 이 지역 위험도"  
예측

### 안전 지대 범위안내

사용자 설정 지역 주변 범위  
파악 후

그 범위의 안전 지대 안내

### 대상 확장

청소년 범죄, 노인 대상 범죄  
예방 정보까지 포함

여성 귀갓길 안전 서비스

# 6. 자체 평가 및 의견

## 염지원

이번 2차 프로젝트는 1차 때보다 규모가 훨씬 커지고, 해야 할 일도 많았다. 특히 데이터베이스 구조와 관련해 변경 사항이 많아, 이를 적용하고 테스트하는 과정에서 어려움이 컸다. 또한, 개발을 진행하며 머지 과정에서 충돌이 발생하는 경우가 많았고, 이를 해결하는 것이 가장 힘들었던 부분이었다.

첫 팀장을 맡아 부족한 점도 많았지만, 팀원들이 잘 따라주고 협력해 준 덕분에 프로젝트를 무사히 마무리할 수 있었던 것 같다.

## 박세원

2차 프로젝트에서 DB를 구성하는 과정이 예상보다 어려웠다. 처음 기획 단계에서 필요하다고 생각한 속성들이 막상 개발 과정에서는 사용되지 않는 경우가 있었고, 반대로 코드를 작성하면서 새로 추가해야 하는 속성들이 많이 생겼다.

이로 인해 DB 구조가 자주 변경되면서 혼란이 발생했다. 다음 프로젝트에서는 초기 설계 단계에서 DB 구조를 더 꼼꼼하게 계획해야겠다고 느꼈다.

## 이치우

팀장님, 팀원님들이 다들 잘 해 주셔서 결과물이 기대 이상으로 나온 것 같다.

아쉬운 점은 각자 crud기능을 하나 이상 해야하는 팀 프로젝트의 성격상 어쩔 수 없는 부분이긴 했지만 기능이 크게 다르지 않은 게시판들은 한 사람이 모두 담당 했으면 좋았을 거라는 생각이 들었다. 스프링부트를 쓰지 않아 스프링 시큐리티 설정에서 생각보다 너무 많은 시간을 소비해 일정에 차질이 생기고 쪽지 기능이 다소 빈약하게 구현된 것도 조금 아쉬웠다.

## 정지현

이번 프로젝트는 1차 때보다 규모가 훨씬 커진 만큼 준비 과정과 진행 과정 모두에서 쉽지 않은 점이 많았다. 기획 단계에서부터 세부 조율이 더 많이 필요했고, 예기치 못한 변수에도 대응해야 했다.

또한 여러 인원이 같이 하는 만큼 같은 페이지를 수정하는 경우가 있어 병합할 때 충돌나는 경우가 많아 오류 수정에 시간을 많이 보낸 게 아쉬웠던 것 같다.



# 감사합니다

