

A wide-angle photograph of a classic movie theater. The foreground shows the dark, silhouetted heads of a large audience seated in rows of seats. The middle ground features a massive, brightly lit white screen centered in the frame. On either side of the screen are large, floor-to-ceiling red velvet curtains. The walls of the theater are made of light-colored stone or plaster, and there are small circular decorative elements on the wall between the curtains. The overall atmosphere is that of a grand, historical cinema.

Cinemu

영화 예매 사이트 구현

Contents

1. 프로젝트 개요

1-1. 기획의도

1-2. Plan

Cinemu

Cinemu 기획 의도

우리가 개발하고자 하는 영화 사이트는 사용자들이 보고 싶은 영화를 쉽게 찾을 수 있도록 다양한 추천 시스템을 구현하여 넷플릭스의 장점과 기존 영화 사이트의 장점을 합쳤습니다.

이를 통해 예매율을 높여 수익을 증대시키는 것이 목표입니다. 또한, 영화 정보 제공, 예매 시스템 구현, 편의 기능 제공 등 다양한 기능을 제공하여 사용자들이 영화를 더욱 편리하게 즐길 수 있도록 하고자 합니다.

Cinemu



Plan

1차 팀프로젝트

2차 팀프로젝트

final 팀프로젝트

01.

1차 팀프로젝트

2022.12.22 ~ 2022.12.28

설계 및 기획

02.

2차 팀프로젝트

2023.01.27 ~ 2023.02.09

UI 작업 및 Spring CRUD

03.

Final 팀 프로젝트

2023.03.06 ~ 2023.04.03

빅데이터 기반 핵심기능 구현

Contents

2. 진행 프로세스

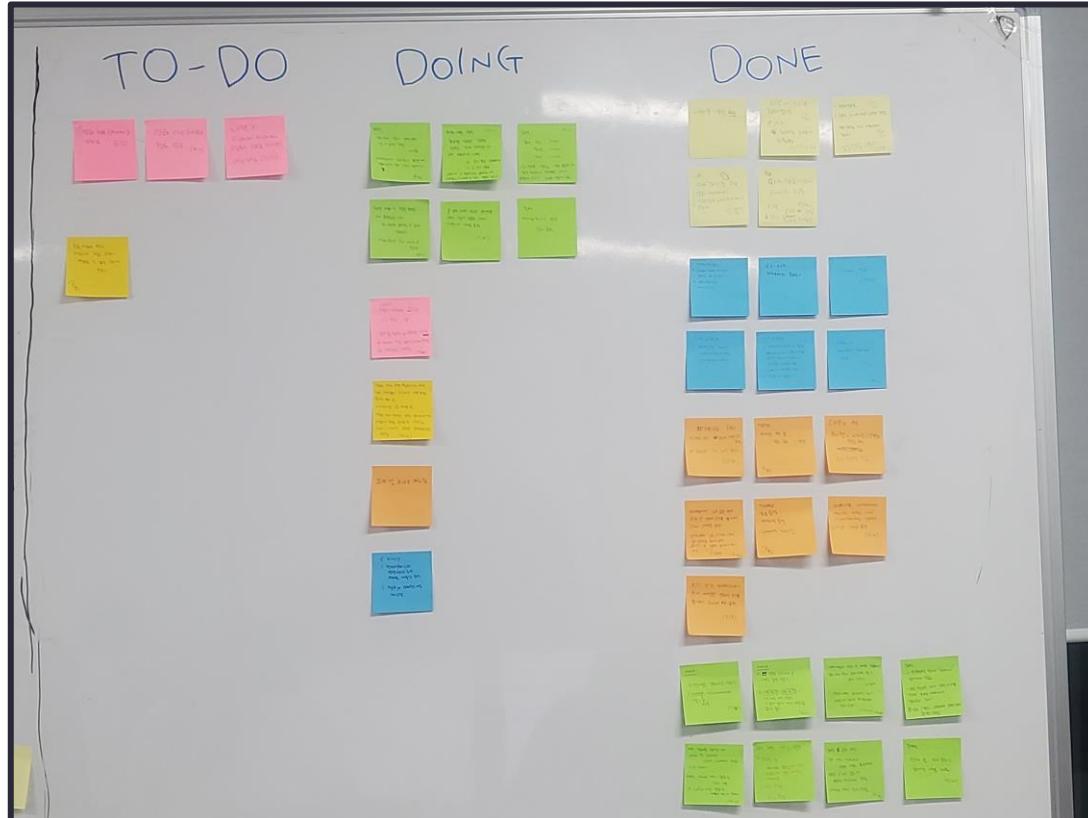
2-1. 개발 일정

2-2. 개발 환경

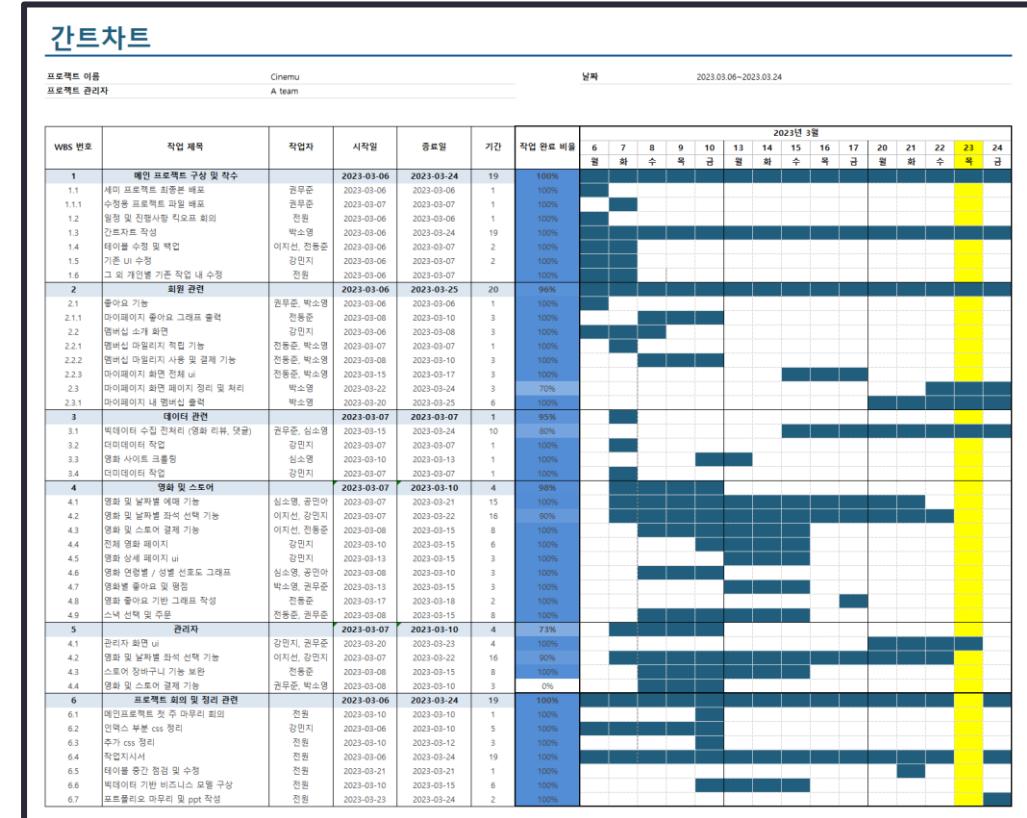
Cinemu

2-1. 개발일정

칸반 보드



간트 차트



100...%

2-2. 개발 환경 - 개발 도구 및 사용기술



TensorFlow



KoNLPy

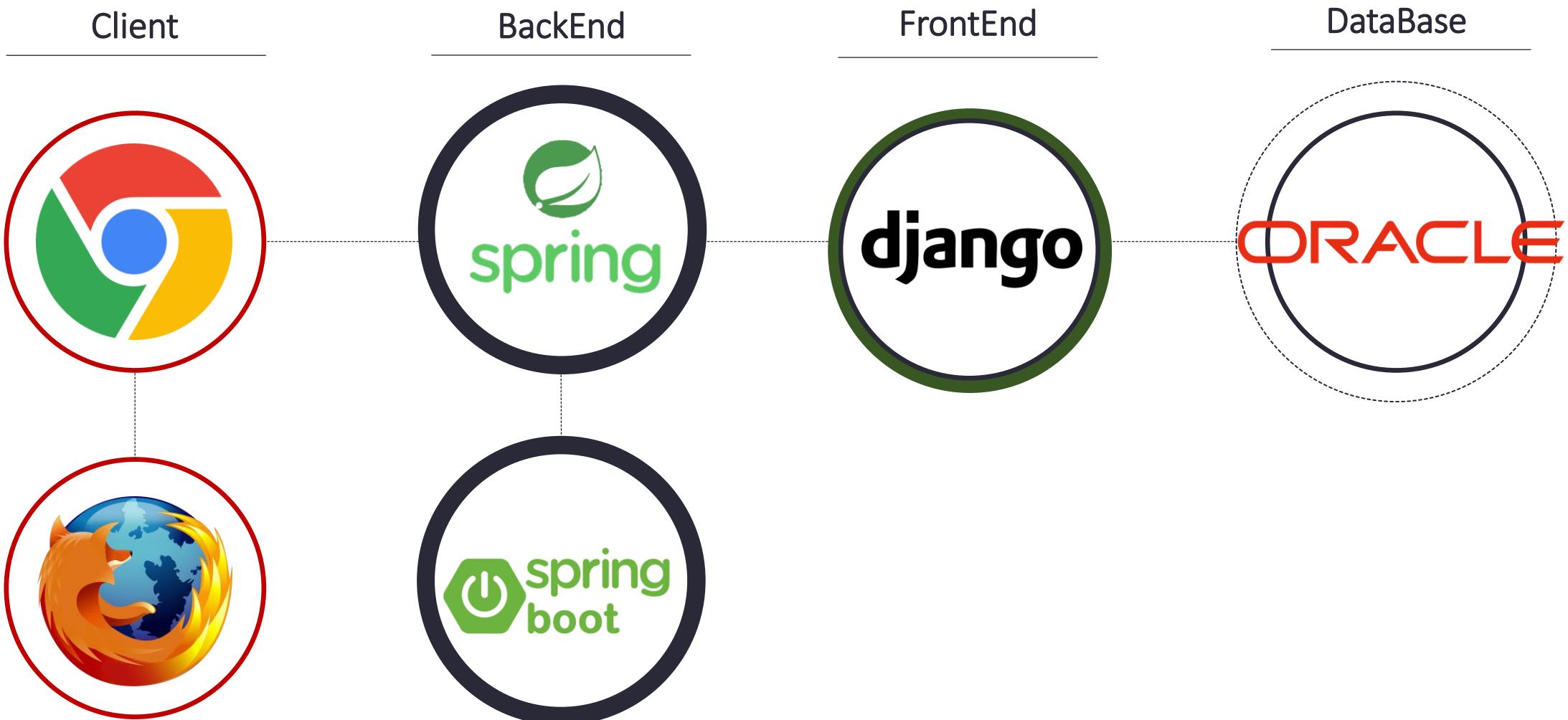


Visual Studio Code



Visual Paradigm





Contents

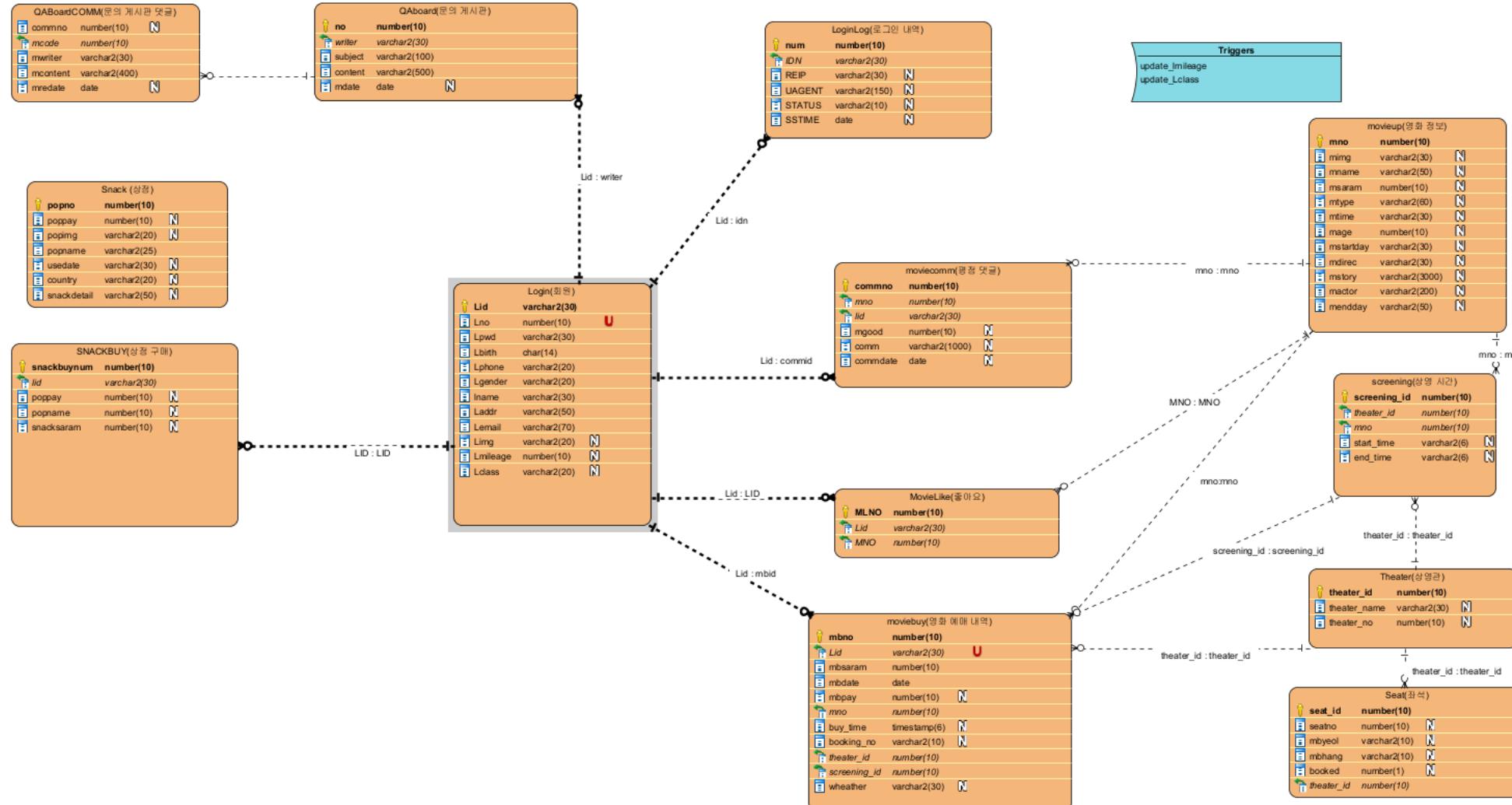
3. 프로젝트 상세

3-1. ERD

3-2. 린 캔버스

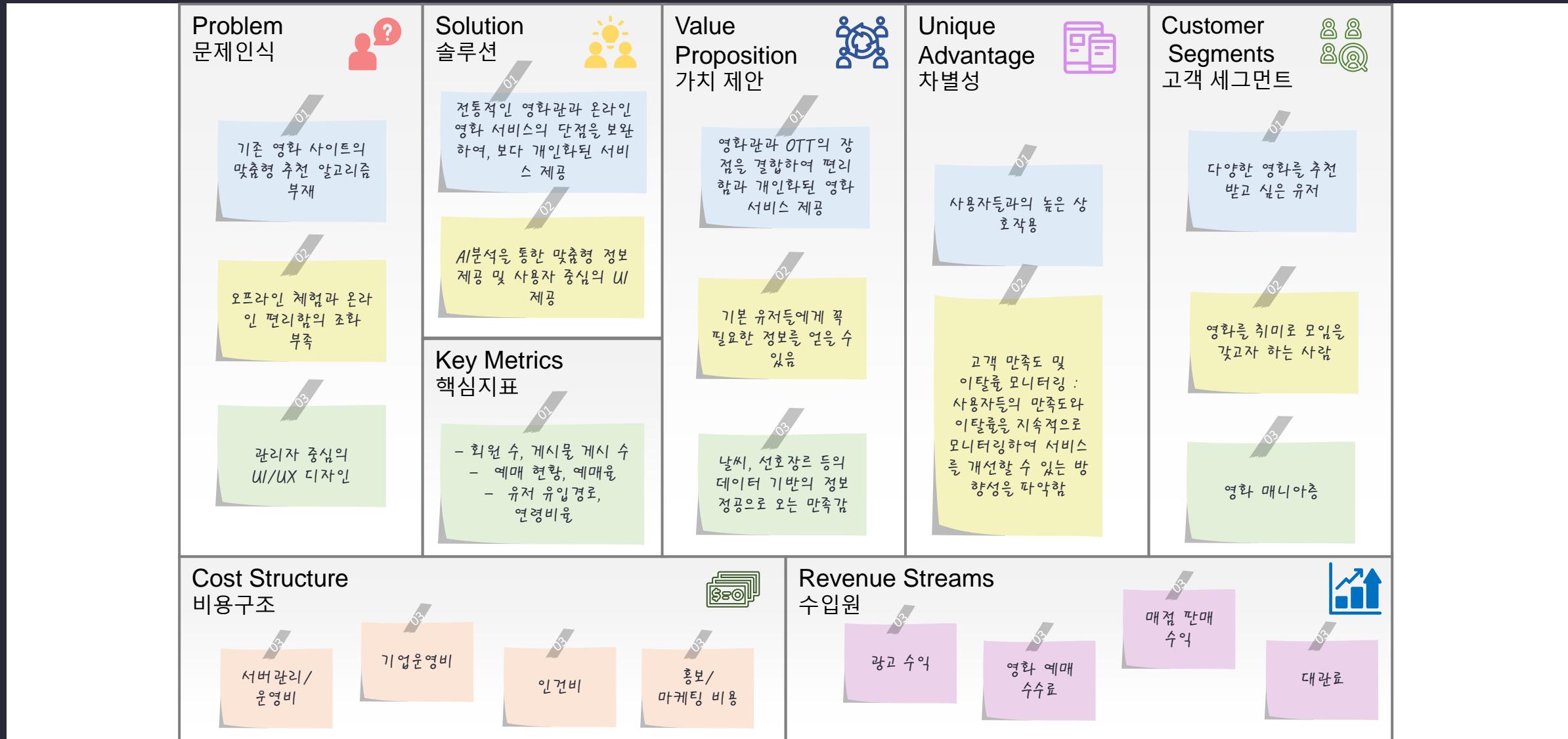
Cinemu

3-1. ERD (Entity-Relationship Diagram)



3-2. 린 캔버스

프로젝트명 : 영화 예매 사이트 구현



Contents

4. 기술 상세

- 4-1. 로그인 및 회원가입
- 4-2. 마이페이지
- 4-3. 영화 리스트
- 4-4. 영화 상세페이지
- 4-5. 예매 및 좌석 선택
- 4-6. 스토어 및 결제
- 4-7. Q&A 게시판
- 4-8. 데이터 수집
- 4-9. 전체 검색
- 4-10. 관리자 페이지
- 4-11. API
- 4-12. 추천시스템

영화 예매 사이트 구현

Cinemu

A team Project

4-1. 로그인 및 회원가입

회원가입

The screenshot shows the 'SIGN UP' form on the Cinemu website. The user has entered their information: ID (cinemu), Name (씨네무), Password (*****), Confirmation Password (*****), Phone Number (010-1111-1111), and Email (cinemu@naver.com). A red box highlights the '비밀번호 재확인' (Password Confirmation) field, which contains '일치합니다' (Matched). Another red box highlights the '중복 확인' (Duplicate Check) button, which is labeled '사용 가능' (Available). A large red arrow points from the '중복 확인' button to the '비밀번호 재확인' field.

```
// 아이디 중복 확인
@RequestMapping(value = "/idchk")
public ModelAndView idChk(String lid) {
    ModelAndView mav = new ModelAndView("loginJoin/idchk");
    int cnt = loginDaoInter.idCheck(lid);
    mav.addObject("cnt", cnt);
    return mav;
}

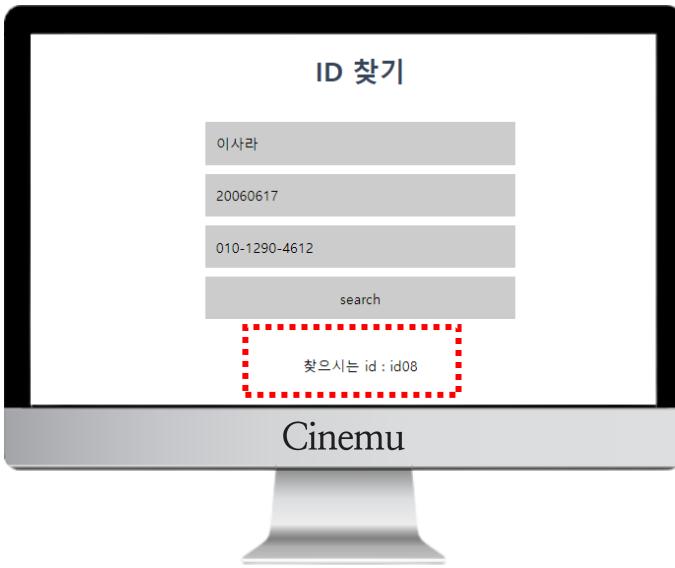
// 비밀번호 이중 확인
function pwdChk() {
    var pw1 = $('#pw1').val();
    var pw2 = $('#pw2').val();
    if (pw1 === pw2) {
        $('#pwchk').css('color', 'green').html('일치합니다');
    } else { // 0
        $('#pwchk').css('color', 'red').html('일치하지 않습니다');
    }
};
```

로그인

The screenshot shows the 'Login' page on the Cinemu website. It features two input fields: 'ID' and 'Password'. Below these is a dark blue 'Login' button. At the bottom of the page, there is a link '아직 멤버가 아닌가요? 지금 멤버되려가기' (Not a member yet? Join now) and links for '아이디 찾기' (Find ID) and '비밀번호 찾기' (Find Password). The word 'Cinemu' is visible at the bottom of the monitor.

```
// 로그인
@PostMapping(value = "/login")
public ModelAndView logIn(HttpServletRequest session, HttpServletResponse request, LoginDTO vo,
    @RequestHeader("User-Agent") String userAgent) {
    ModelAndView mav = new ModelAndView("redirect:/main");
    LoginDTO dto = loginDaoInter.login(vo);
    if (dto == null) {
        mav.setViewName("error/paramException");
        mav.addObject("emsg", "로그인 실패입니다.");
    } else {
        session.setAttribute("sessionName", dto.getLname());
        session.setAttribute("sessionId", dto.getLid());
    }
    return mav;
}
```

- ID 찾기 - id가 있을 때



```

@PostMapping(value = "/searchId")
public ModelAndView searchId(LoginDTO vo) {
    ModelAndView mav = new ModelAndView("loginJoin/findId");
    String id = loginDaoInter.findById(vo);
    if (id != null) {
        mav.addObject("fid", 1);
        mav.addObject("id", id);
        System.out.println("id2" + id);
    } else {
        mav.addObject("fid", 0);
    }
    return mav;
}

```

- ID 찾기 - id가 없을 때



```

<form action="searchId" method="post">
    <h1>ID 찾기</h1>
    <input type="text" placeholder="이름" name="lname" /> <input
        type="text" placeholder="생년월일 YYYYMMDD ex) 850101" name="lbirth" />
    <input type="text" placeholder="전화번호 ex) 010-0000-0000"
        name="lphone" /> <input type="submit" value="search" />
    <p>
        <c:choose>
            <c:when test="${fid == 1}">
                찾으시는 id : ${id}
            </c:when>
            <c:when test="${fid == 0}">
                정보가 없습니다
            </c:when>
            <c:otherwise>
                <p>정확한 정보를 입력해주세요</p>
            </c:otherwise>
        </c:choose>
    </p>
</form>

```

4-1. 로그인 및 회원가입

- 비밀번호 찾기 품



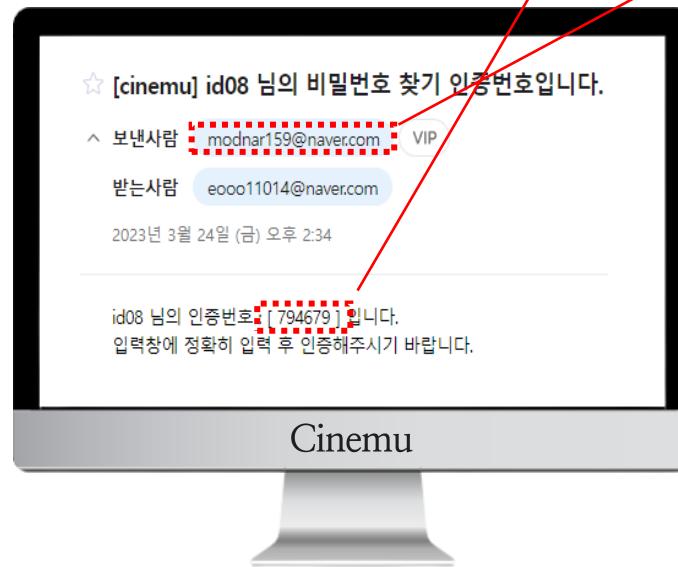
```
// 비밀번호 찾기 시, 인증번호 발송
@PostMapping(value = "/sendRnum")
public ModelAndView sendRnum(LoginDTO vo, HttpServletRequest request, HttpServletResponse response) {

    String email = (String) request.getParameter("lemail");
    String id = (String) request.getParameter("lid");
    LoginDTO v = loginDaoInter.findPwd(vo);
    String lid = v.getLid();
    Random r = new Random();
    int rnum = r.nextInt(999999); // Random 클래스를 사용하여 999999까지의 랜덤 난수 생성으로 인증번호 발송

    if (v.getLid().equals(id)) {
        System.out.println("173번 줄 : " + id);
        if (v.getEmail().equals(email)) {
            String emailToRecipient = v.getEmail();
            String emailSubject = "[cinemu] " + lid + " 님의 비밀번호 찾기 인증번호입니다.";
            String emailMessage = lid + " 님의 인증번호는 [" + rnum + "] 입니다." + "\n" + "입력창에 정확히 입력 후 인증해주시기 바랍니다.";
            String emailFromRecipient = "modnar159@naver.com";

            try {
                mailSender.send(new MimeMessagePreparator() {
                    public void prepare(MimeMessage mimeMessage) throws Exception {
                        MimeMessageHelper mimeMsgHelperObj = new MimeMessageHelper(mimeMessage, true, "UTF-8");
                        mimeMsgHelperObj.setTo(emailToRecipient);
                        mimeMsgHelperObj.setFrom(emailFromRecipient);
                        mimeMsgHelperObj.setText(emailMessage);
                        mimeMsgHelperObj.setSubject(emailSubject);
                    }
                });
            } catch (Exception e) {
                ModelAndView mav = new ModelAndView("loginJoin/findPwd");
                String error = "가입하신 id 또는 email의 정보를 찾을 수 없습니다";
                mav.addObject("error", error);
            }
        }
    }
    else {
        ModelAndView mav = new ModelAndView("loginJoin/findPwd");
        String error = "가입하신 id 또는 email의 정보를 찾을 수 없습니다";
        mav.addObject("error", error);
        return mav;
    }
    else {
        ModelAndView mav = new ModelAndView("loginJoin/findPwd");
        String error = "가입하신 id 또는 email의 정보를 찾을 수 없습니다";
        mav.addObject("error", error);
        return mav;
    }
}
}
```

- 인증번호 확인 메일



- Mail 발송을 위한 Bean 설정

```
<!-- mailing -->
<bean id="mailSender"
      class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.naver.com" />
    <property name="port" value="465" />
    <property name="username" value="modnar159@naver.com" />
    <property name="password" value="kmj225000" />
    <property name="defaultEncoding" value="utf-8" />
    <property name="javaMailProperties">
        <prop key="mail.debug">true</prop>
        <prop key="mail.smtp.starttls.enable">true</prop>
        <prop key="mail.smtps.ssl.checkserveridentity">true</prop>
        <prop key="mail.smtp.auth">true</prop>
        <prop key="mail.smtps.ssl.trust">*</prop>
        <prop key="mail.smtp.ssl.protocols">TLSv1.2</prop>
        <prop key="mail.smtp.socketFactory.class">
            javax.net.ssl.SSLSocketFactory
        </prop>
    </props>
</property>
</bean>
```

4-1. 로그인 및 회원가입

- 인증번호 체크 : 이메일로 받은 인증번호 (랜덤 난수 rnum)를 Ajax를 사용하여 체크.



```
// 인증번호 체크 메서드
@RequestMapping(value = "/chkRnum", produces = "application/json;charset=utf-8")
public int chkRnum(int rnum, String inputrnum) {
    int inputr = Integer.parseInt(inputrnum);
    if(rnum == inputr) {
        return 1;
    } else {
        return 0;
    }
}
```

인증번호 체크 Ajax 부분

```
$('#rnumCheckForm').submit(function(event) {
    event.preventDefault();
    var form = $('#rnumCheckForm');
    var data = $('#rnumCheckForm').serialize();
    $.ajax({
        url : form.attr('action'),
        data : data,
        dataType : "json",
        error : function(e) {
        },
        success : function(jsondata) {
            if (jsondata == 1) {
                $('#result').html('인증번호가 일치합니다');
                $('#moveUpPwd').show();
            } else {
                $('#result').html('인증번호가 일치하지 않습니다');
            }
        }
});
```

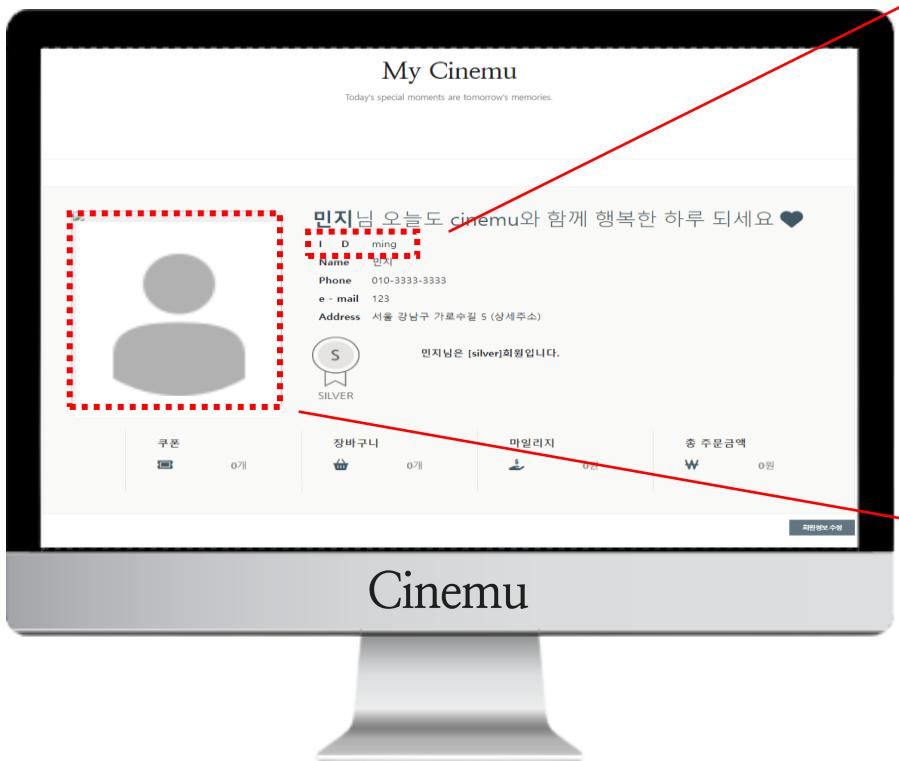
```
// 비번 변경
@RequestMapping(value = "/upPwd")
public ModelAndView upPwd(String lid, String lpwd) {
    ModelAndView mav = new ModelAndView("redirect:/member/loginForm");
    Map<String, String> map = new HashMap();
    map.put("lid", lid);
    map.put("lpwd", lpwd);
    loginDaoInter.upPwd(map);
    mav.addObject("map", map);
    return mav;
}
```

- 비밀번호 재설정 : 인증번호 일치 시, 비밀번호 재설정

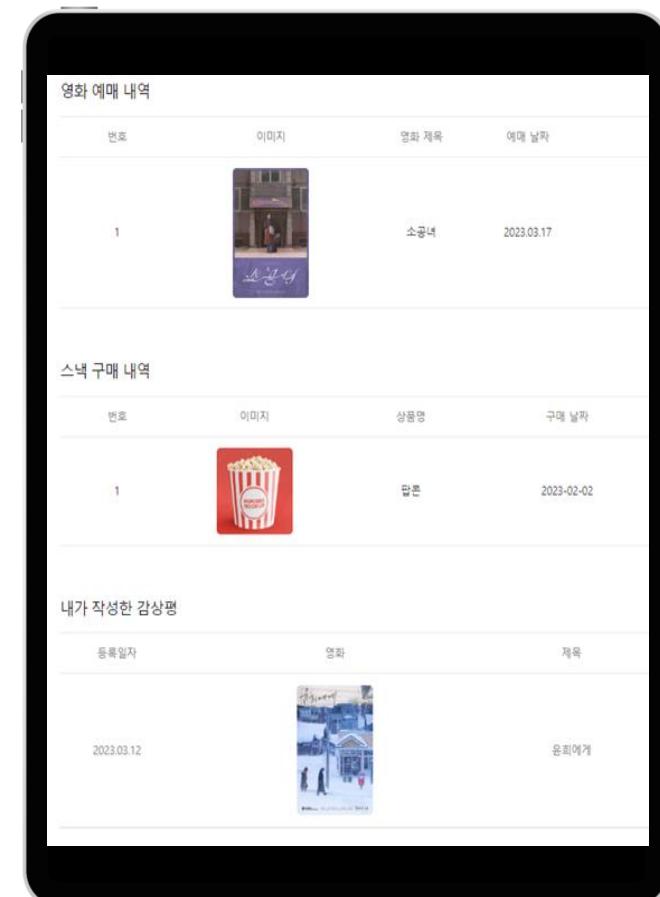


4-2. 마이페이지

- **마이페이지** : 로그인 후 session을 이용해서 자신의 정보를 확인하는 페이지 - 'My Cinemu'
회원가입 후에는 지정해 둔 기본 포토로 처음 등록되며, 마일리지와 등급 확인이 가능.
File 객체를 생성하여 프로필 사진 수정, 삭제가 가능.



```
// mypage
@GetMapping(value = "/mypage")
public ModelAndView mypage(HttpServletRequest request) {
    HttpSession session = request.getSession(false);
    String id = (String) session.getAttribute("sessionID");
    ModelAndView mav = new ModelAndView();
    if(id != null) {
        LoginDTO v = loginDaoInter.mypage(id);
        List<MovieUpDTO> muvo = myPageInter.movieP(id);
        List<SnackUpDTO> snvo = myPageInter.snackP(id);
        List<LoginDTO> lovo = myPageInter.qnaP(id);
        request.setAttribute("v", v);
        request.setAttribute("muvo", muvo);
        request.setAttribute("snvo", snvo);
        request.setAttribute("lovo", lovo);
        mav.setViewName("loginJoin/myPage");
    } else {
        mav.setViewName("index");
    }
    return mav;
}
// 프로필 사진 수정
@RequestMapping(value = "/photoup")
public String upload(Model m, LoginDTO vo, HttpServletRequest request) {
    String img_path= "resources\\imgpwd"; //이미지 경로
    String r_path = request.getRealPath("/");
    String oriFn = vo.getMfile().getOriginalFilename(); // 파일 이름
    long size = vo.getMfile().getSize(); //용량
    String contentType = vo.getMfile().getContentType(); // 타입
    StringBuffer path = new StringBuffer(); // 임시 저장소
    path.append(r_path).append(img_path).append("\\");
    path.append(oriFn);
    File f = new File(path.toString()); // 파일 객체 생성
    try {
        vo.getMfile().transferTo(f);
        vo.setImg(oriFn);
    } catch (IllegalStateException | IOException e) {
        e.printStackTrace();
    }
    loginDaoInter.update(vo);
    return "redirect:/member/mypage";
}
```



4-2. 마이페이지

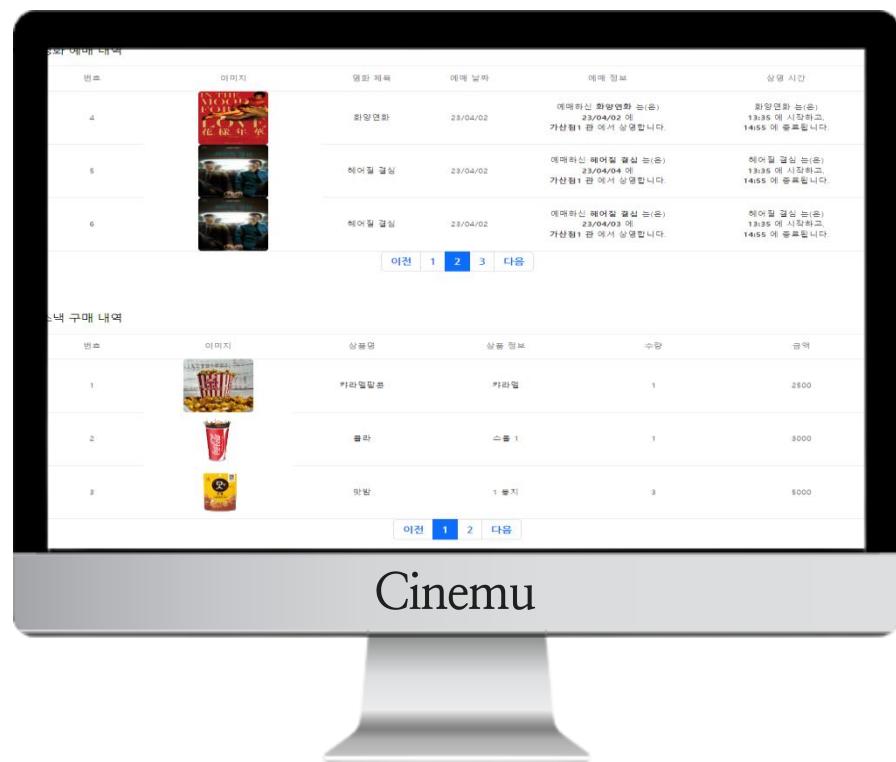
• 마이페이지 – 개인 활동 내역 :

로그인 후 마이페이지에서 활동 내역을 확인할 수 있다.

예매내역, 구매내역, 문의글, 좋아요 영화 리스트, 영화 감상평 출력.

한 페이지에 출력하기 위하여 restController를 이용, Ajax로 내역 출력 및 페이징.

* 구현 로직이 같으므로, 예매 내역 코드를 첨부.



1. Mapper

join을 하여 mapper 작성. resultMap 대신 myvo DTO를 별도로 작성 후 resultType으로 지정. 날짜형식을 substr으로 정리하여 통일화함.

```
<select id="movieip" parameterType="map" resultType="myvo">
    SELECT rnum, mimg, mname, mbdate, buy_time, seatno, theater_name,
    theater_no, start_time, end_time
    FROM (
        SELECT rownum rnum,
        mu.mimg, mu.mname, SUBSTR(mb.mbdate, 1, 10) mbdate, SUBSTR(mb.buy_time, 1, 8) buy_time,
        mb.seatno,
        th.theater_name, th.theater_no, sc.start_time, sc.end_time
        FROM movieup mu
        JOIN moviebuy mb ON mu.mno = mb.mno
        JOIN theater th ON mb.theater_id = th.theater_id
        JOIN screening sc ON mb.screening_id = sc.screening_id
        WHERE mb.lid = #{lid}
    )
    WHERE rnum BETWEEN #{begin} and #{end}
</select>

<select id="movieCnt" parameterType="String" resultType="int">
    SELECT count(mname) FROM movieup
    mu, moviebuy mb, theater th, screening sc where mu.mno = mb.mno
    and
    mb.theater_id = th.theater_id
    and mb.screening_id = sc.screening_id
    and
    mb.lid =#{lid}
</select>
```

2. Config

MyMovieDTO / alias="myvo" 를 작성하여 Mapper에 이용.

```
<typeAlias type="movie.pak.dto.MyMovieDTO" alias="myvo" />
```

4-2. 마이페이지

- **마이페이지 – restController를 이용, Ajax로 내역 출력 및 페이지.**
 - * 구현 로직이 같으므로, 예매 내역 코드를 첨부.

1. restController : Map으로 id, begin, end 값을 전달해서 Ajax 페이지처리.

```
@RestController
@RequestMapping(value = "/memberR")
public class RestControllerMember {

    private int perPage=3; // 한 페이지 당 게시물 수
    private int totalCount; // 총 게시물 수
    private int totalPage ; // 총 페이지 수

    @Autowired
    private LoginDAOInter loginDaoInter;

    @Autowired
    private MyPageDAOInter myPageInter;

    @RequestMapping(value = "/mvpage", produces = "application/json;charset=utf-8")
    public Map<String, Object> mvpage(HttpServletRequest request){
        HttpSession session = request.getSession(false);
        String id = (String) session.getAttribute("sessionID");
        String cPage = request.getParameter("cPage");
        Map<String, Object> map = new HashMap<>();
        int begin = (Integer.parseInt(cPage) - 1) * 3 + 1;
        int end = Integer.parseInt(cPage) * 3;
        map.put("lid", id);
        map.put("begin", begin);
        map.put("end", end);

        List<MyMovieDTO> list = myPageInter.movieP(map);
        totalCount = myPageInter.movieCnt(id);
        totalPage = (int) Math.ceil(totalCount / (double) perPage); // 총 페이지 수
        Map<String, Object> pmap = new HashMap<>();
        pmap.put("list", list);
        pmap.put("cPage", cPage);
        pmap.put("mvprevBtn", (Integer.parseInt(cPage) > 3));
        pmap.put("mvnextBtn", (Integer.parseInt(cPage) < totalPage));
        pmap.put("totalPage", totalPage); // 총 페이지 수
        pmap.put("totalCount", totalCount); // 총 페이지 수
        return pmap;
    }
}
```

- 2. JSP : 총 5개 내역을 Ajax로 전송하고, 5개 ajax function을 호출하여 각각 페이지되도록 구현.

```
$function() {
    mvcomm();
    qna();
    snack();
    movie();
    mvlike();

    function mvlike() {
        let uid ='<%=String session.getAttribute("sessionID")%>';
        let perPage = 3;
        let totalCount = 0;
        let totalPage = 0;
        let totalBlock; // 전체 블럭수
        let beginPerPage; // 각 페이지별 시작 게시물의 index값
        let endPerPage; // 각 페이지별 마지막 게시물의 index값
        let pagePerBlock = 3; // 한 블럭당 보여질 페이지 수
        let nowBlock = 1; // 현재 블럭 -> [][][][] -> 1block당 5개 배열 생성
        // 처음 페이지 로딩 시 1페이지를 보여줌
        getmvl(1);

        function getmvl(cPage) {
            $.ajax({
                url: "${pageContext.request.contextPath}/memberR/mvlpage?cPage=" + cPage,
                success: function(data) {
                    let table = $("#mvitable");
                    table.empty();
                    for (let i = 0; i < data.list.length; i++) {
                        let tr = $("<tr>");
                        tr.append($("<td>").text(data.list[i].rnum));
                        tr.append($("<img>").attr("src", "${pageContext.request.contextPath}/resources/img/semi_movie/" + data.list[i].mimg));
                        tr.append($("<td>").text(data.list[i].mname));
                        tr.append($("<td>").text(data.list[i].mtype));
                        table.append(tr);
                    }
                    // 페이지 처리
                    totalPage = data.totalPage; // totalPage 변수 초기화
                    totalBlock = Math.ceil(totalPage / pagePerBlock); // 전체 블록 수 계산
                    nowBlock = Math.ceil(cPage / pagePerBlock); // 현재 블록 계산
                    beginPerPage = (nowBlock - 1) * pagePerBlock + 1; // 현재 페이지 범위 내에서 보여질 시작 페이지 번호 계산
                    endPerPage = nowBlock * pagePerBlock; // 현재 페이지 범위 내에서 보여질 끝 페이지 번호 계산
                    endPerPage = (endPerPage > totalPage) ? totalPage : endPerPage; // 현재 페이지 범위 내에서 보여질 끝 페이지 번호 조정

                    var pagination = $(".mvlpagination");
                    pagination.empty();
                    // 이전 블럭 버튼 생성
                    pagination.append("<li class='page-item' id='mvlpagelink'><a class='mvlpagelink' href='#'>이전</a></li>");
                    // 페이지 버튼 생성
                    for (var i = beginPerPage; i <= endPerPage; i++) {
                        var active = "";
                        if (i == cPage) {
                            active = "active";
                        }
                        var page = "<li class='page-item' " + active + "><a class='mvlpagelink' href='#" + onclick='getmvl(' + i + ");'>" + i + "</a></li>";
                        pagination.append(page);
                    }
                    pagination.append("<li class='page-item' id='mvlnextlink'><a class='mvlpagelink' href='#'>다음</a></li>");
                },
            });
        }
    }
}
```

4-2. 마이페이지



ORACLE

```
create or replace TRIGGER update_lclass
BEFORE INSERT ON moviebuy
FOR EACH ROW
DECLARE
    total_pay NUMBER;
    grade VARCHAR2(10);
BEGIN
    SELECT SUM(mbpay) INTO total_pay
    FROM moviebuy
    WHERE lid = :new.lid;

    IF total_pay > 500000 THEN
        grade := 'diamond';
    ELSIF total_pay > 300000 THEN
        grade := 'platinum';
    ELSIF total_pay > 100000 THEN
        grade := 'gold';
    ELSE
        grade := 'silver';
    END IF;

    UPDATE login
    SET lclass = grade
    WHERE lid = :new.lid;
END;
```

Trigger

"moviebuy" 테이블에 insert 되기 전에 실행됩니다.
본인 아이디에서 구매한 금액을 모두 합하여 그 금액에 따라 마일리지 등급을 결정하고, 등급 조건의 값이 넘었을 때 해당 id를 가진 유저의 등급을 바꿔 줍니다.

ORACLE

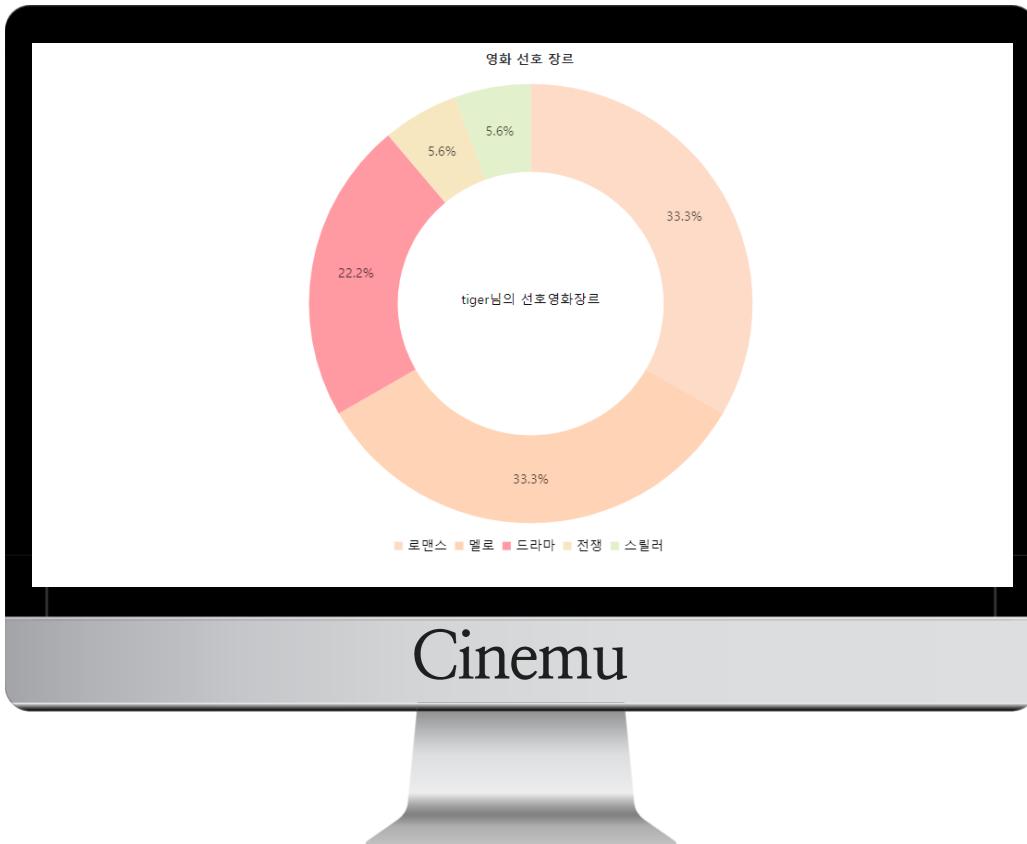
```
create or replace TRIGGER update_lmileage
AFTER INSERT ON moviebuy
FOR EACH ROW
DECLARE
    mileage_rate NUMBER(3,2);
BEGIN
    SELECT CASE lclass
        WHEN 'silver' THEN 0.01
        WHEN 'gold' THEN 0.03
        WHEN 'platinum' THEN 0.05
        WHEN 'diamond' THEN 0.1
        ELSE 0 END
        INTO mileage_rate
        FROM login
        WHERE lid = :new.lid;

    UPDATE login
    SET lmileage = lmileage +
    (:new.mbpay * mileage_rate)
    WHERE lid = :new.lid;
END;
/
```

Trigger

"moviebuy" 테이블에 insert 될 때마다 실행됩니다.
"login" 테이블에서 등급에 따라 마일리지 비율을 결정하고, 해당 id를 가진 유저의 마일리지 값을 증가시킵니다.

사용자 선호 영화 장르 차트



Spring Controller

Arrays.asList() 사용하여 /로 구분된 장르를 나누어 리스트로 변환한 후 for문으로 리스트를 돌며 map에 장르와 그에 해당하는 카운트 값을 저장.

reverseOrder()로 역순 정렬을 해준 후 Collectors 클래스의 toMap 메소드를 이용하여 key, value를 그대로 유지하며, 동일한 key가 있는 경우 값을 더하여 병합하고, 최종 결과를 LinkedHashMap으로 반환

```

@RestController
public class ChartRestController {
    @Autowired // 경계와
    public ChartDAOInter chartDAOInter;

    @Autowired
    private MyPageDAOInter myPageDaoInter;

    @RequestMapping(value = "/likemtype", produces = "application/json;charset=utf-8")
    public String test(String lid) {
        List<String> mtype = myPageDaoInter.likemtype(lid);
        Map<String, Integer> map = new HashMap<>();

        for (String type : mtype) {
            List<String> mtypeList = Arrays.asList(type.split("/")); // Arrays.asList() 사용하여 /로 구분된 문자열을 나누어 리스트로 변환
            for (String genre : mtypeList) {
                Integer count = map.getOrDefault(genre, 0);
                Integer allcount = count + 1;
                map.put(genre, allcount); // 리스트를 돌며 map에 장르와 그에 해당하는 카운트 값을 저장
            }
        }

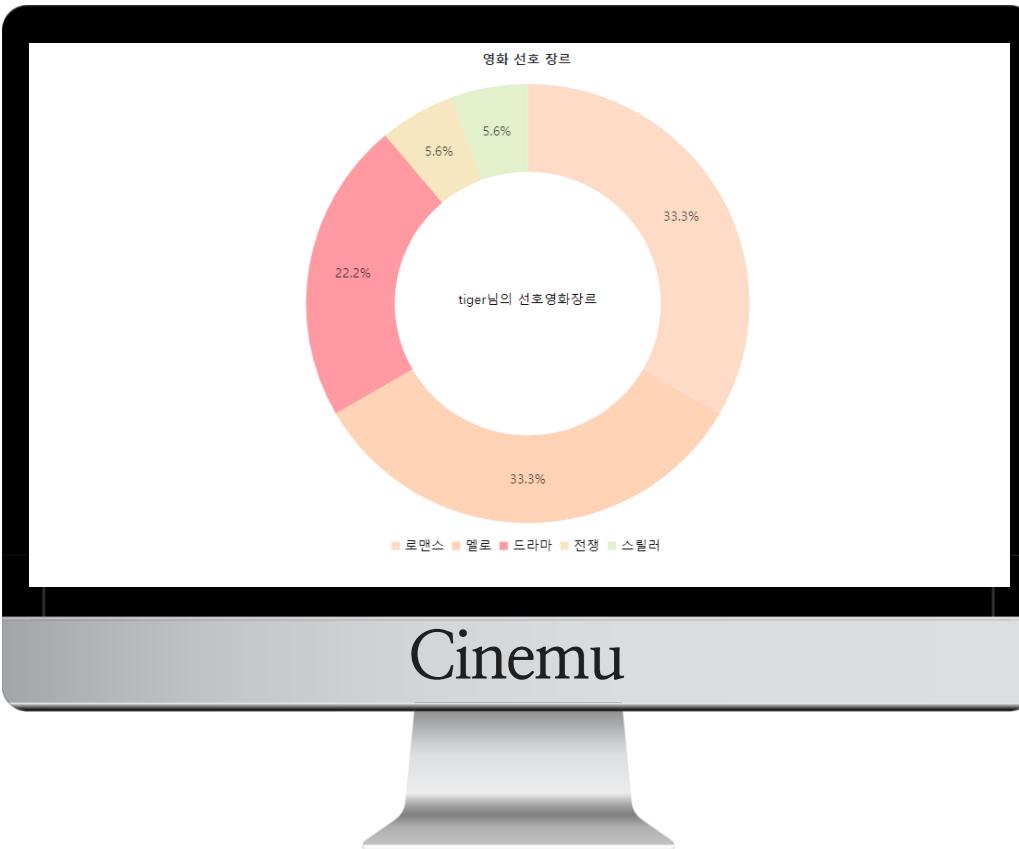
        Map<String, Integer> sortedMap = map.entrySet().stream()
            .sorted(Map.Entry.comparingByValue(Comparator.reverseOrder())) // 역순정렬
            .limit(5) // 것수는 5개까지로 정해줌
            .collect(Collectors.toMap( // Collectors 클래스의 toMap 메소드를 이용하여 key를 그대로 유지하고, value를 그대로 유지하며, 동일한 key가 있는 경우 값을 더하여 병합하고,
                // 최종 결과를 LinkedHashMap으로 변환
                Map.Entry::getKey, Map.Entry::getValue, (e1, e2) -> e1, LinkedHashMap::new)); // LinkedHashMap은 입력된 순서를 유지 -> 아래의 주석 코드와 같음
        /* Map<String, Integer> map = new LinkedHashMap<>();
        for (Map.Entry<String, Integer> entry : entries) {
            map.put(entry.getKey(), entry.getValue());
        }
        */

        String ref = null;
        ObjectMapper mapper = new ObjectMapper(); // ObjectMapper를 사용하여 Map 객체를 JSON 형태로 변환
        try {
            ref = mapper.writeValueAsString(sortedMap); // 변환한 후, ref 문자열에 저장
            ref = "[" + ref + "]"; // "[{},"역선":27,"드라마":13,"스릴러":11,"로맨스":10,"코미디":8}]" -> 이런식으로 만들
            System.out.println(ref);
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
        return ref;
    }
}

```



사용자 선호 영화 장르 차트



Ajax, c3 chart

Session으로 로그인한 아이디를 받아 온 후
Ajax와 c3 도넛 차트를 이용하여 최종 사용자별 선호 장르 출력

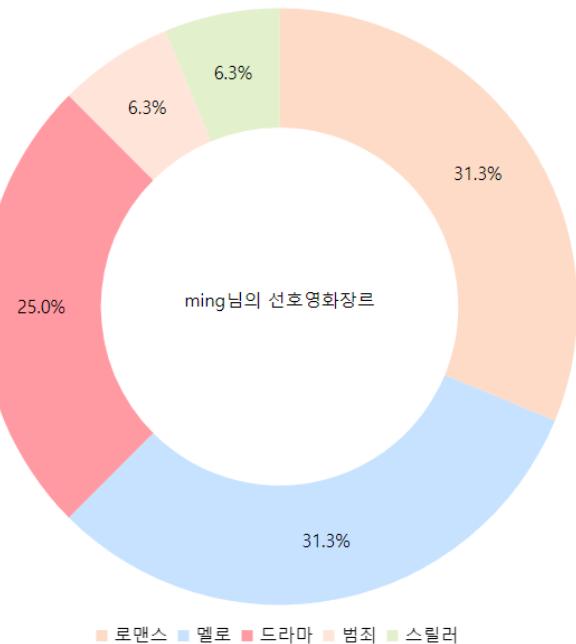
```

var uid = '<%=String(session.getAttribute("sessionID"))%>';
console.log("uid====>"+uid);
$(function() {
$.ajax({
    url : "./likemtype?lid="+uid,
    success : function(jsondata) {
        console.log(jsondata)
        var chart = c3.generate({
            bindto : '#pie',
            size: {
                width: 800,
                height: 600
            },
            data : {
                json : [jsondata[1]],
                keys : {
                    value : Object.keys(jsondata[1]),
                },
                type : 'donut',
                colors: {
                    '드라마': '#FF9AA2',
                    '엘로': '#FFD385',
                    '로맨스': '#DDBC77',
                    '미스터리': '#D6EFFF',
                    '판타지': '#A0C4FF',
                    '다큐멘터리': '#B5EAD7',
                    'SF': '#E2F0CB',
                    '스릴러': '#FFF1C1',
                    '코미디': '#FB782',
                    '애니메이션': '#FD95E5',
                    '액션': '#CAE4DB',
                    '서사': '#EDB183',
                    '공포': '#C5DCA0',
                    '느와르': '#B2B2B2',
                    '가족': '#E4F9F5',
                    '범죄': '#FFE5D9',
                    '서스펜스': '#B7B7A4',
                    '전쟁': '#F6E7C1',
                    '모험': '#FFC2E2'
                }
            },
            donut : {
                title : uid+'님의 선호영화장르',
            }
        });
    }
});
```

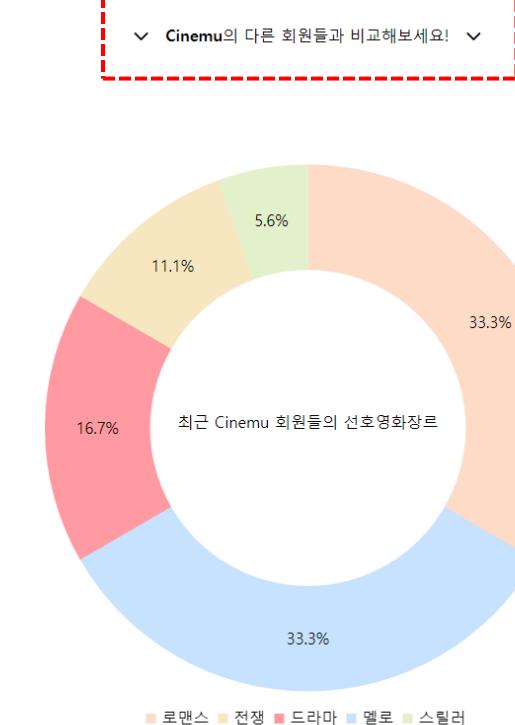


4-2. 마이페이지

사용자 선호 장르 및 영화 이름



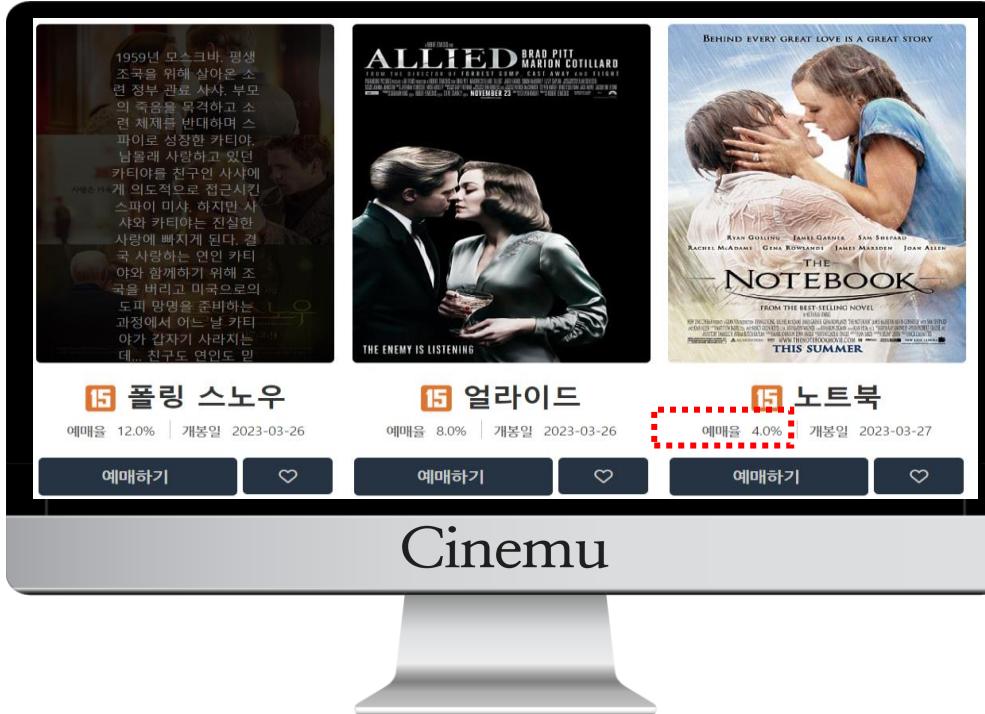
Cinemu 선호 장르 및 영화 이름



兀라이드
스윗
프랑세즈
스윗
프랑세즈
스윗
프랑세즈
풀링 스노우
헤어질 결심
노트북
색, 계
화양연화
스윗
프랑세즈
노트북
풀링 스노우

장르 클릭 시 사용자 선호 장르 및 해당 장르 별 영화와 Cinemu 사용자의 선호 장르 및 대항 별 영화를 비교하여 볼 수 있음

4-3. 영화 리스트



jsp

```
<c:forEach var="e" items="${list}" varStatus="status">
  <strong class="percent"><a href="#">예매율</a><span>${ratio[status.index]}%</span></strong>
```

영화 리스트의 forEach 안에 예매율의 forEach문을 사용하고자
varStatus= 'status' 를 사용해서
"ratio"라는 배열에서 "status.index" 위치에 있는 값을 가져옴

예매율



Mapper

```
<!-- 영화 예매율 -->
<select id="ratio" resultType="float">
  SELECT
    NVL(ratio, 0) AS ratio
  FROM movieup mu
  LEFT JOIN (
    SELECT
      MU.mno,
      ROUND(SUM(MB.mbsaram) / (SUM(SUM(MB.mbsaram)) OVER() * 0.01), 2) AS ratio
    FROM movieup MU
    JOIN moviebuy MB ON MU.mno = MB.mno
    WHERE MB.buy_time BETWEEN TRUNC(SYSDATE) - 6 AND TRUNC(SYSDATE) + 1
    GROUP BY MU.mno, MU.mname
  ) mr ON mu.mno = mr.mno
  ORDER BY mu.mno
</select>
```



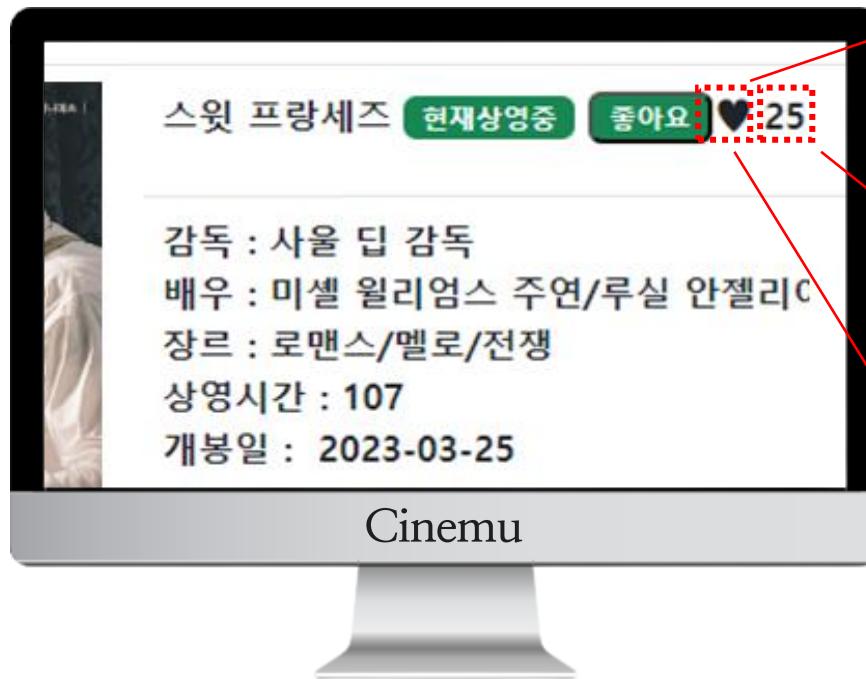
Mapper

최근 7일 동안 구매된 영화의 예매율을 계산하여 조회하는 쿼리.

Left Join으로 movieup" 테이블과 "moviebuy" 테이블을 조인.
movieup" 테이블의 모든 데이터를 가져오면서, "moviebuy" 테이블에서 구매
정보가 있는 경우에만 해당 정보를 가져오도록 설정.
최근 7일 동안 구매된 영화 데이터를 필터링하기 위해 "WHERE" 구문에서
"buy_time"이 7일 이내인 데이터만을 가져옴.
GROUP BY" 구문을 사용하여 영화 번호(mno)를 기준으로 그룹화하고, "SUM"
함수와 "OVER" 함수를 사용하여 총 업로드 용량의 1%를 나누어 계산하여 업
로드 비율을 구함
"ORDER BY" 구문을 사용하여 영화 번호(mno)를 기준으로 오름차순으로 정
렬하여 결과를 반환

4-4. 영화 상세페이지

- **좋아요 기능** : 좋아요 클릭을 하였을 때 insert 되고, 재클릭했을 때 delete 되도록 하여 session 당 한번씩만 inser가 될 수 있도록 함. Ajax를 이용하였고, 하트 총 개수를 구할 수 있음. 마이페이지 내 선호도 그래프와 연동.



1. Mapper

좋아요 수를 반환 : 기록이 있으면 1 없으면 0을 반환하여 중복 '좋아요'를 방지.

```
<select id="myLikeCnt" parameterType="mlvo" resultType="int">
    select count(mlno) from movielike where mno=#{mno} and lid=#{lid}
</select>
```

전체 좋아요 수를 반환

```
<select id="likeCnt" parameterType="mlvo" resultType="int">
    select count(mlno) as likecnt from movielike where mno=#{mno}
</select>
```

2. Controller

```
// 좋아요 클릭 시 실행되는 메서드
@RequestMapping(value = "/addlike", produces = "application/json;charset=utf-8")
public Map Like(MovieLikeDTO mlvo, HttpSession session){
    Map map = new HashMap();
    String id = (String) session.getAttribute("sessionID");

    int cnt=movieLikDaoInter.mylikeCnt(mlvo);
    if(id == null) {
    }else if (cnt==0) {
        movieLikDaoInter.addLike(mlvo);
    }else {
        movieLikDaoInter.delLike(mlvo);
    }
    int cnt2=movieLikDaoInter.mylikeCnt(mlvo);
    int likecnt = movieLikDaoInter.likeCnt(mlvo);
    map.put("cnt2", cnt2);
    map.put("likecnt", likecnt);
    return map;
}

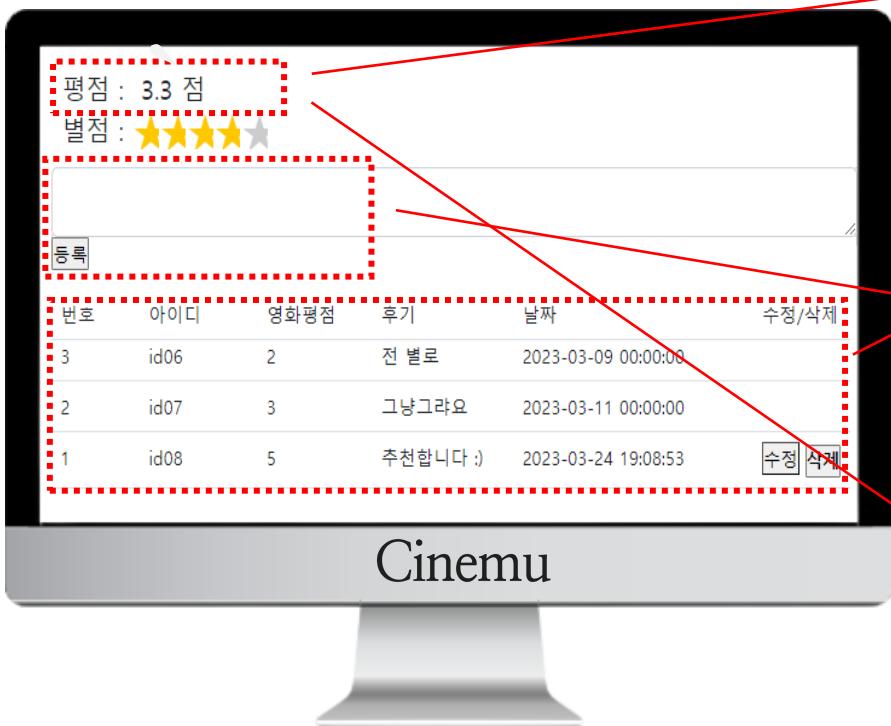
@RequestMapping(value = "/likecnt2", produces = "application/json;charset=utf-8")
public Map likeCnt(MovieLikeDTO mlvo){
    Map map = new HashMap();
    // 추가 카운트 값을 결정 -> 좋아요 기록이 있다면 "비 카운트", / 좋아요 기록이 있다면 "최종 카운트"
    int cnt=movieLikDaoInter.mylikeCnt(mlvo); 전체 좋아요 수
    int likecnt = movieLikDaoInter.likeCnt(mlvo);
    map.put("cnt", cnt);
    map.put("likecnt", likecnt);
    return map;
}
```

3. Ajax

```
$.ajax({
    url : "../like/likecnt2?mno=${vo.mno}&lid=${sessionScope.sessionID}",
    error : function(e) {
    },
    success : function(jsondata){
        if(jsondata.cnt==0){
            $('#heart').html('♡');
        }else{
            $('#heart').html('♥');
        }
        $('#likecnt').html(jsondata.likecnt);
    }
})
```

4-4. 영화 상세페이지

- 평점 및 영화 후기 댓글 : id 로그인 후 후기와 별점을 매길 수 있고, 전체 평점을 출력. 본인이 쓴 댓글과 평점 수정 / 삭제가 가능함.



2. Controller

```
@RequestMapping(value = "/getavg")
public float mgoodAvg(int mno) {
    float avg=movieUpDAOInter.mgoodAvg(mno);
    return avg;
}

@GetMapping(value = "/moviedetail")
public ModelAndView detailmovie(int no) {
    ModelAndView mav = new ModelAndView("movie/detailMovie");
    List<MovieCommDTO> listmoviecomm = movieUpDAOInter.listMovieComm(no);
    mav.addObject("listmoviecomm", listmoviecomm);
    MovieUpDTO vo = movieUpDAOInter.detailMovie(no);
    mav.addObject("vo", vo);
    return mav;
}

@PostMapping(value = "/moviecomminsert")
public ModelAndView addMovieComm(MovieCommDTO mcvo) {
    movieUpDAOInter.addMovieComm(mcvo);
    ModelAndView mav = new ModelAndView("redirect:/movies/moviedetail?no="+mcvo.getMno());
    return mav;
}

@RequestMapping(value = "/upmoviecomm")
public String upMovieComm(MovieCommDTO mcvo) {
    int num = mcvo.getMno();
    movieUpDAOInter.upMovieComm(mcvo);
    return "redirect:/movies/moviedetail?no="+ num;
}
```

3. Ajax

```
$.ajax({
    url:"../movier/getavg?mno="+${vo.mno},
    error: function (xhr, status, error) {
    },
    success: function(json){
        if(json !== null){
            $('#mgoodAvg').html(json+"점");
        }else if(json === null){
            $('#mgoodAvg').html("별점을 등록해주세요 :)");
        }
    });
});
```

1. Mapper

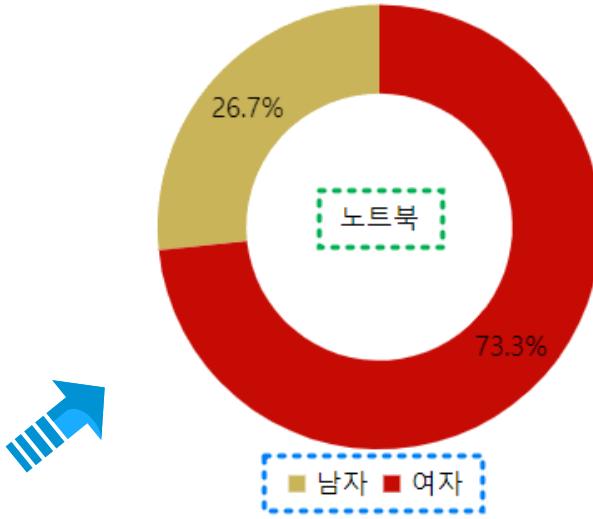
컬럼에 없는 평점을 avg 식을 사용한 쿼리문을 이용하여 작성.

```
<select id="mgoodAvg" parameterType="int" resultType="float">
    select
        round(AVG(NVL(mgood,0)),1) as mgoodAvg from moviecomm where
        mno=#{mno}
</select>
```

4-4. 영화 상세페이지 – 도넛 차트

성별 예매 분포도

성별 예매 분포도



RestController

```
@RequestMapping(value = "/genDonut",
    produces = "application/json;charset=utf-8")
public String genChart(int mno) {
    String mvTitle = chartDAOInter.movieTitle(mno);
    List<LoginDTO> list = chartDAOInter.genDonut(mno);
    Map<String, Integer> map = new HashMap<>();

    for (int i = 0; i < list.size(); i++) {
        map.put(list.get(i).getGender(), list.get(i).getCnt());
    }
    String res = null;
    ObjectMapper mapper = new ObjectMapper();
    try {
        res = mapper.writeValueAsString(map);
        res = "[" + "{\"movieTitle\":\"" + mvTitle + "\"}, " + res + "]";
    } catch (JsonProcessingException e) {
        e.printStackTrace();
    }
    return res;
}
```

JSP

```
$.ajax({
    url : "../genDonut?mno=${vo.mno}",
    success : function(jsondata){
        var chart = c3.generate({
            bindto : "#genDonut",
            data : {
                json : [jsondata[1]],
                keys : {
                    value : Object.keys(jsondata[1])
                },
                type : "donut",
                colors: {
                    '여자' : '#C60B05',
                    '남자' : '#c9b459'
                }
            },
            donut : {
                title : jsondata[0].movieTitle,
            }
        });
        //-----
    },
    error : function(e) {
        console.log("실패")
    }
});
```



설명

- Ajax를 이용하여 응답받은 데이터를 페이지 이동 없이 출력한다.
- list에 저장된 LoginDTO 객체의 성별과 관람객 수를 map에 저장한다.
- ObjectMapper를 사용하여 Map 객체를 JSON 형태로 변환한다.
- try문을 통해 출력된 결과는 아래와 같다.

0번지 : {movieTitle: '노트북'}
1번지: {남자: 4, 여자: 11}

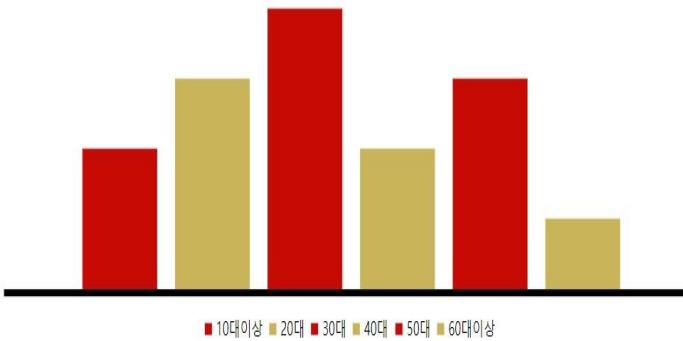
설명

- jQuery로 Ajax 요청을 보낸다.
- JSON 형식의 데이터를 c3.js 라이브러리를 사용하여 도넛 차트로 출력한다.
- Key를 통해 1번지에 저장된 성별을 출력한다.

4-4. 영화 상세페이지 – 바 차트

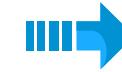
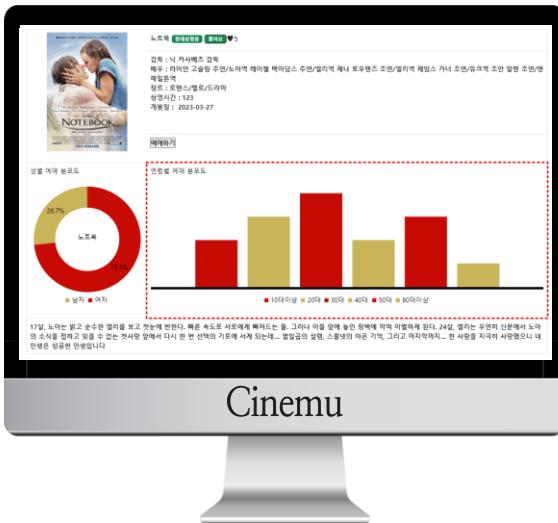
연령별 예매 분포도

연령별 예매 분포도



Mapper

```
<select id="ageBarlist" parameterType="int" resultType="map">
    select ages, cnt from
    (SELECT
        sum(decode(age2,'10s',1,0))a10,
        sum(decode(age2,'20s',1,0))a20,
        sum(decode(age2,'30s',1,0))a30,
        sum(decode(age2,'40s',1,0))a40,
        sum(decode(age2,'50s',1,0))a50,
        sum(decode(age2,'60s',1,0))a60
    FROM (SELECT
        <![CDATA[
        CASE WHEN age < 20 THEN '10s'
        WHEN age >= 20 AND age < 30 THEN '20s'
        WHEN age >= 30 AND age < 40 THEN '30s'
        WHEN age >= 40 AND age < 50 THEN '40s'
        WHEN age >= 50 AND age < 60 THEN '50s'
        WHEN age >= 60 THEN '60s' ]]>
        END age2
    FROM (
        SELECT trunc(months_between(sysdate,lbirth)/ 12) age
        from login l, movieup m, moviebuy b
        where b.lid = l.lid(+) and m.mno=b.mno and b.mno = #{mno})) )
    UNPIVOT
    cnt for ages in (a10 as '10대이상',a20 as '20대',a30 as '30대',a40 as '40대',a50
    as '50대',a60 as '60대이상'))
</select>
```



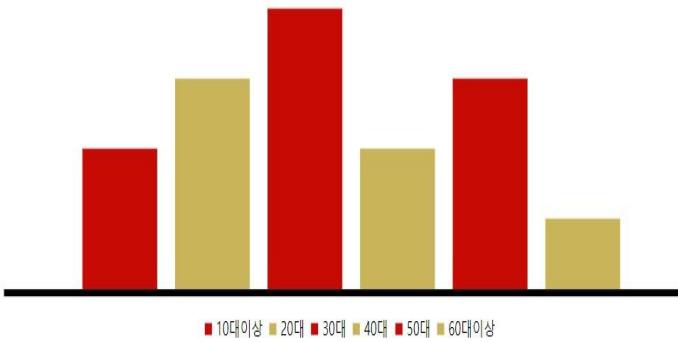
설명

- CASE WHEN을 사용하여, 구매한 사용자들의 나이를 age2 변수로 저장한다.
- trunc를 사용하여 개월 수를 12개월(1년)로 나눈 값의 소수점 이하는 제거한다.
- UNPIVOT을 사용하여 열을 행으로 변환한다.
- <![CDATA 를 사용하여 쿼리문에 부등호를 사용한다.

4-4. 영화 상세페이지 – 바 차트

연령별 예매 분포도

연령별 예매 분포도



RestController

```
@RequestMapping(value = "/ageBar",
    produces = "application/json;charset=utf-8")
public String ageChart(int mno) {
    String mvTitle = chartDAOInter.mvTitle(mno);
    List<Map<String, Integer>> list = chartDAOInter.ageBarlist(mno);
    Map<Object, Object> map = new LinkedHashMap<Object, Object>();
```



설명

- LinkedHashMap을 이용하여 put에서 입력된 순서대로 key와 value를 저장한다.

JSP

```
$.ajax({
    url : ".../ageBar?mno=${vo.mno}",
    success : function(jdata) {
        var chart = c3.generate({
            bindto : "#agelist",
            data : {
                labels: Object.keys(jdata[1])[0],
                json : [jdata[1]],
                keys : {
                    value : Object.keys(jdata[1])
                },
                type : 'bar',
                colors:{
```

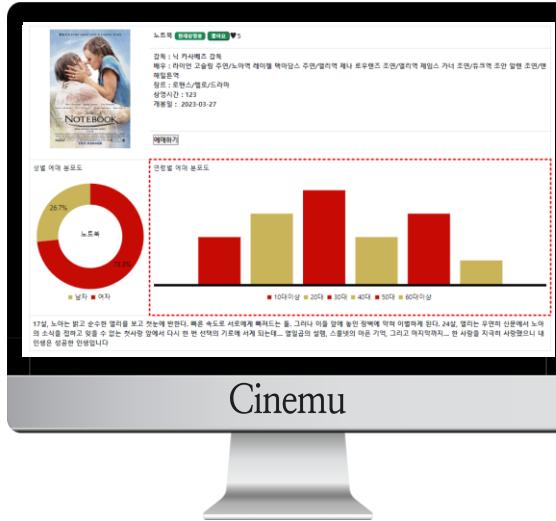
'10대이상': '#C60B05',
'20대': '#c9b459',
'30대': '#C60B05',
'40대': '#c9b459',
'50대': '#C60B05',
'60대이상': '#c9b459'

```
            },
            axis: {
                x: {
                    type: 'categories',
                    categories: ['']
                },
                y: {
                    show:false
                }
            },
        });
    }
});
```



설명

- JSON 형식의 데이터를 c3.js 라이브러리를 사용하여 바차트로 출력한다.



극장별, 날짜별 출력

Mapper

2023 04

3 일	4 일	5 일	6 일	7 일
월요일	화요일	수요일	목요일	금요일

 스위프팅 콘서트 11:00 ~ 13:00 13:00 ~ 15:00
 스위프팅 콘서트 13:00 ~ 15:00 11:00 ~ 13:00 14:00 ~ 16:00



```

<select id="daysList" parameterType="hashmap" resultMap="muvoreresult">
    select mp.mname, mp.mage, sc.start_time, sc.end_time, sc.theater_id from movieup mp, screening sc, theater th
    where
    <choose>
        <when test="theater_name != null and mendday == null">
            mp.mno=sc.mno and sc.theater_id = th.theater_id and theater_name=#{theater_name}
        </when>
        <when test="theater_name != null and mendday != null">
            mp.mno=sc.mno and sc.theater_id = th.theater_id and
            <![CDATA[<mendday>=#{mendday}</mendday>]]>and theater_name=#{theater_name}
        </when>
        <otherwise>mp.mno=sc.mno and sc.theater_id = th.theater_id </otherwise>
    </choose>
    </where>
</select>

<resultMap id="muvoreresult" type="muvo">
    <result column="mno" property="mno"/>
    <result column="mname" property="mname"/>
    <result column="mage" property="mage"/>
    <collection resultMap="timevoretult" property="screenList" ofType="timevo" javaType="java.util.List">
    </collection>
</resultMap>

<resultMap id="timevoretult" type="timevo">
    <result column="start_time" property="start_time"/>
    <result column="end_time" property="end_time"/>
    <result column="theater_id" property="theater_id"/>
</resultMap>

```



설명

- choose-when-otherwise문
동적 쿼리 사용
- theater_name과 mendday에 따라
상영리스트 출력값에 변화를 준다.
- resultMap으로 muvoreresult
timevoretult 를 이용하여, column과
property를 각각 매핑한다.

가로 달력, 작은 달력 연동

Jsp

Cinemu Schedule

극장별
가산점
잠실점
용산점

2023. 04

3 일	4 일	5 일	6 일	7 일	8 일	9 일	10 일	11 일	12 일	13 일	14 일	15 일	16 일	17 일
월요일	화요일	수요일	목요일	금요일	토요일	일요일	월요일	화요일	수요일	목요일	금요일	토요일	일요일	월요일

장편영
15 어버웃타임
15:00 ~ 17:00

18 본즈 편 을
13:00 ~ 15:00




```

$( '#datevalue' ).change(function() {
    let testval = $(this).val();
    let changeval = testval.substr(-2);
    if(changeval[0] == '0') {
        changeval = changeval.substr(-1);
    }

    let testem = $('button.dayclick-btn').children('em').filter(function() {
        return $(this).text() == changeval;
    });

    $('button.dayclick-btn').children('em').css("background-color", "transparent");
    $('button.dayclick-btn').css("background-color", "transparent");

    $('button.dayclick-btn').children('em').filter(function() {
        return $(this).text() == changeval;
    }).trigger('click');
    $('button.dayclick-btn').children('em').filter(function() {
        return $(this).text() == changeval;
    });
});

```



설명

- substr(-2)을 사용하여 \$('#datevalue')에서 가져온 값에서 마지막 두 글자를 추출한다.
- filter() 함수를 사용하여 선택된 일자 중에서 changeval과 일치하는 일자를 선택한다.
- trigger() 메소드를 통해 해당 em 엘리먼트에 등록된 클릭 이벤트 핸들러가 매번 실행된다.

4-5. 예매 및 좌석 선택 – 상영시간표(2)

가로 달력, 작은 달력 연동

Jsp

Cinemu Schedule

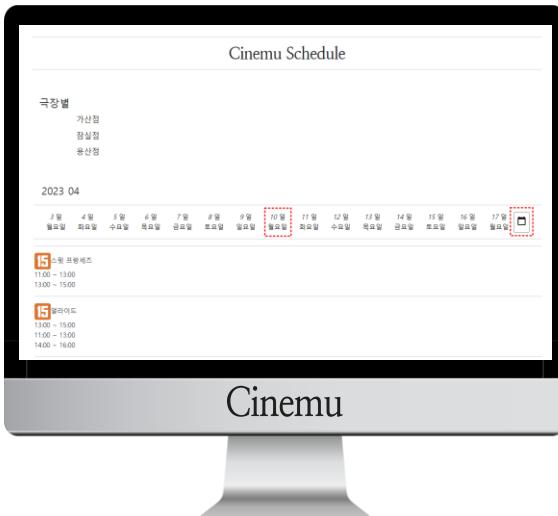
극장별
가산점
잠실점
용산점

2023. 04

3 일	4 일	5 일	6 일	7 일	8 일	9 일	10 일	11 일	12 일	13 일	14 일	15 일	16 일	17 일
월요일	화요일	수요일	목요일	금요일	토요일	일요일	월요일	화요일	수요일	목요일	금요일	토요일	일요일	월요일

장편영
15 어버웃타임
15:00 ~ 17:00

18 본즈 편 을
13:00 ~ 15:00



```
$('.dayclick-btn').click(function() {
    if (theater_name === null) {
        alert("극장을 먼저 선택해주세요.");
    }
}

let day = $(this).children('em').text().padStart(2, '0');
let year = $('.year').text();
let month = $('#monthval').text().trim();
let getDt = year + "-" + month + "-" + day;

$.ajax({
    url : "${pageContext.request.contextPath}/moviebooking/moviedaya?mendday=" +getDt+ "&theater_name=" +theater_name,
    error : function(e) {
        console.log("에이잭스 error");
    },
    success : function(json) {
        let str = '';
        $.each(json, function(k, v){
            str += "<hr><li id='listline'><img id='imgsize' src=${pageContext.request.contextPath}/resources/img/newaging/" +v.mage+ ".png">" +v.mname+ "</li>";

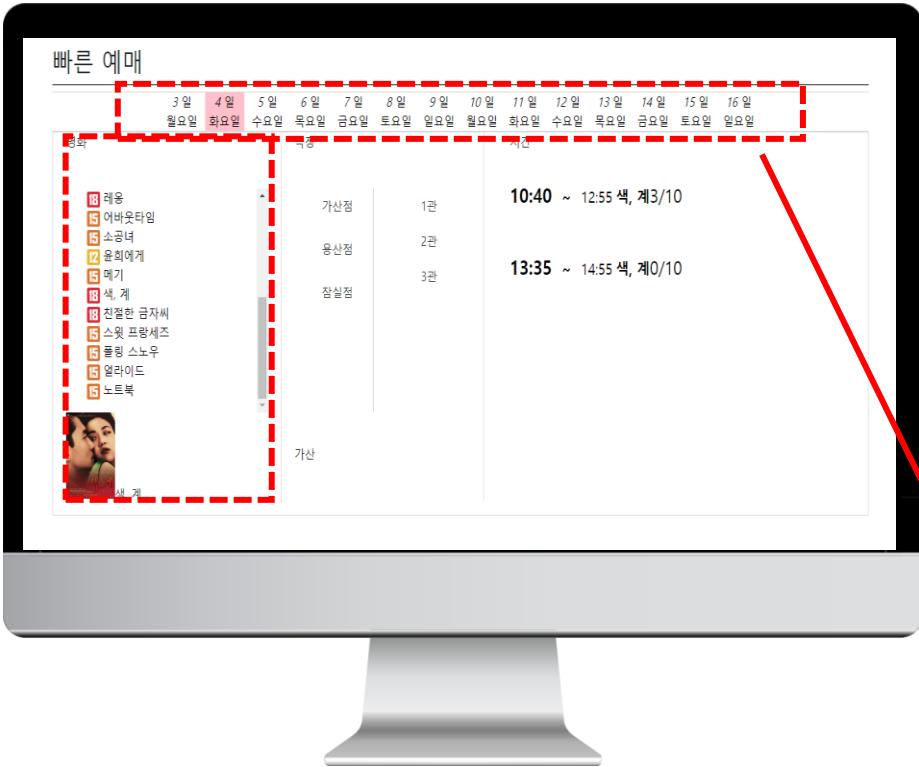
            $.each(v.screenlist, function(k, v) {
                str += "<li id='mstarttime'>" +v.start_time+ " ~ " +v.end_time+ "</li>";
            });
        });
        $("#mCSB_1_container").children('ul').html(str);
    }
});
```

설명

- padStart() 함수를 사용하여 날짜가 한 자리 수인 경우 날짜 앞에 0을 붙여서 2자리로 만든다.
- 인덱스인 k와 value인 v를 사용하여 mage, mname를 \$.each() 통해 순차적으로 출력한다.

4-5. 예매 및 좌석 선택

예매 - 영화 리스트 출력



설명

Mapper <where> 사용 두가지 조건 사용하여 <when> 조건이 성립하지 않아도 mapper은 성립

설명

Ajax를 이용하여 화면에 먼저 영화 리스트 출력

설명

날짜 클릭 했을 때 상영일자에 맞게 리스트 구현

```
<!-- 예매 / 기본 영화리스트 상영일자별 영화 뽑기-->
<select id="daysmovie" parameterType="String" resultType="muvo">
    select mimg,mname,mage from movieup
    <where> |
        <choose>
            <when test="mendday != null">
                <![CDATA[replace(mendday, '-', '') =#{mendday}]]&gt;
            &lt;/when&gt;
        &lt;/choose&gt;
    &lt;/where&gt;
&lt;/select&gt;

//페이지 진입 시 바로 실행되는 ajax

$.ajax({//url에 꼭 테스트해보기
    url : "${pageContext.request.contextPath}/moviebooking/movieday",
    error : function(e) {
        console.log("error");
    },
    success : function(json) {
        let str = '';
        for(const e in json){
            str += '&lt;li&gt;&lt;span style="display: none;"&gt;' + json[e]['mage'] + '&lt;/span&gt;&lt;button class="movie-btn"&gt;&lt;span class="yimg"&gt;&lt;img style="width:20px; height:20px;" src=${pageContext.request.contextPath}/resources/img/newaging/' + json[e]['mage'] + '.png&lt;/span&gt;' + json[e]['mname'] + '&lt;/button&gt;&lt;span style="display: none; class="imgclass"&gt;' + json[e]['mimg'] + '&lt;/span&gt;&lt;/li&gt;';
        }
        $("#mCSB_1_container").children('ul').html(str);
    }
});

$('button.movie-btn').click(function(){
    let imgname = $(this).next('span').text(); //이미지 이름
    let age = $(this).prev().text();
    let moviename = $(this).text(); // 무비 네임
    let pmimg = '&lt;img id="imgid" style="width: 150px; margin-top: 12%; margin-right: 5%; src=${pageContext.request.contextPath}/resources/img/semi_movie/' + imgname + '&gt;';
    let htmlTag = "&lt;p&gt;" + moviename + "&lt;/p&gt;";

    //ajax 시작
    $.ajax({//url에 꼭 테스트해보기
        url : "${pageContext.request.contextPath}/moviebooking/movieday?mendday=" + getDt,
        error : function(e) {
            console.log("error");
        },
        success : function(json) {</pre>
```

4-5. 예매 및 좌석 선택

```
let adultPrice = 15000; // 성인 가격
let teenPrice = 10000; // 청소년 가격
let discountPrice = 5000; // 우대 가격

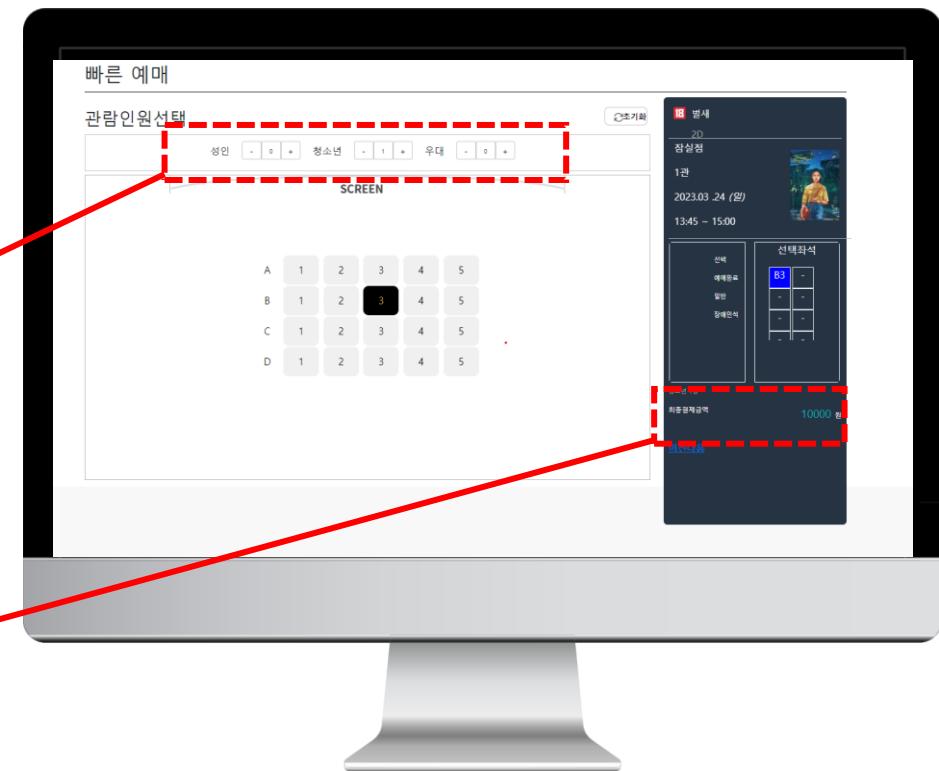
// 연령에 맞게 화면에 띄워지게 하는 함수
function peopleCount() {
let adultCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("성인")')
    .siblings('.count').children('.number').children(".now").text());
let teenCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("청소년")')
    .siblings('.count').children('.number').children(".now").text());
let discountCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("우대")')
    .siblings('.count').children('.number').children(".now").text());

let adultText = "성인 " + adultCount + "명";
let teenText = "청소년 " + teenCount + "명";
let discountText = "우대 " + discountCount + "명";
let result = "";
if (adultCount > 0) {
    result += adultText;
}if (teenCount > 0) {
    if (result !== "") {
        result += ", ";
    }result += teenText;
}if (discountCount > 0) {
    let oldDocument = '우대요금으로 예매하신 고객님께서는 상영관입장시 증빙서류를 제시해 주시기 바랍니다. (미 지참시 입장제한)';
    if ($('p.dot-list').text() !== oldDocument) {
        $('p.dot-list').text(oldDocument);
        $('#document-modal').css('display', 'flex');
    } if (result !== "") {
        result += ", ";
    } result += discountText;
}$("#p.count span").text(result);
}

// 인원 수에 따라 가격을 계산하는 함수
function calculatePrice() {
let adultCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("성인")')
    .siblings('.count').children('.number').children(".now").text());
console.log("성인 요금"+adultCount);
let teenCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("청소년")')
    .siblings('.count').children('.number').children(".now").text());
console.log("청소년 요금"+teenCount);
let discountCount = parseInt($(".check-people-box").children('.check-people').children('.txt:contains("우대")')
    .siblings('.count').children('.number').children(".now").text());
console.log("우대 요금"+discountCount)
// 연령 별 금액
console.log("성인가격"+adultPrice);
// let test = $('.count')
let totalPrice = (adultCount * adultPrice) + (teenCount * teenPrice) + (discountCount * discountPrice);
$(".money em").text(totalPrice); // 가격을 화면에 표시
}

return totalPrice;
}
```

좌석 선택 – 인원 수



설명

연령별 좌석 선택 시 각각 다르게 가격 출력되게 구현

설명

인원 수 감소, 증가 버튼 클릭 시 계산되어 화면에 출력

4-5. 예매 및 좌석 선택

```
// 인원 수에 따라 선택 가능한 좌석 업데이트
function updateSelectableSeats() {
let totalCount = getTotalCount(); // 현재 선택된 인원 수

// 좌석 버튼 비활성화 & 활성화
$('.seatBtn').each(function() {
let seatval = $(this).val();
let seatno = $(this).data('seatno');

if ($(this).hasClass('booked')) {
$(this).prop('disabled', true); // 예약된 좌석은 선택 불가능하게 설정
} else {
if (totalCount <= maxCount) {
$(this).prop('disabled', false);
} else {
if ($(this).hasClass('selected')) {
$(this).prop('disabled', false);
} else {
$(this).prop('disabled', true);
}
}
}
});
calculatePrice(); // 총 가격 계산
}
```

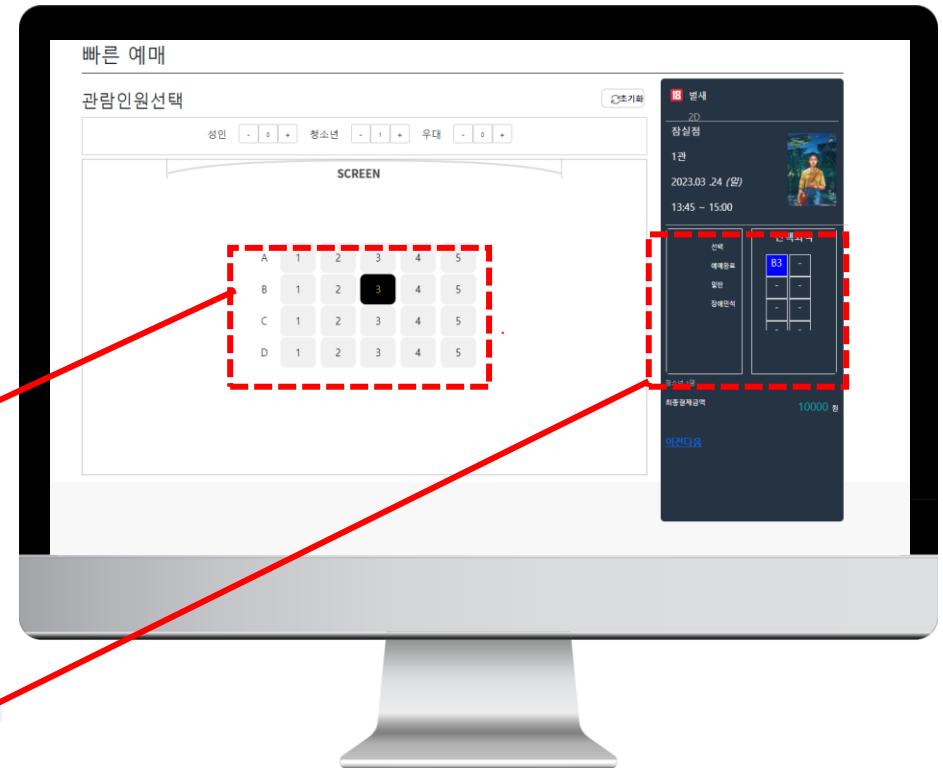
설명

인원 수 만큼 좌석 선택 가능 구현

설명

좌석 활성화&비활성화 기능

좌석 선택 – 좌석 버튼 기능



설명

좌석 버튼 클릭 시 색깔 변화

설명

선택좌석 칸에 선택한 좌석 값이 출력 구현

```
// 선택된 좌석 번호를 화면에 표시
$(this).css('background-color', 'white');
$(this).css('color', 'black');
} else {
// 선택 가능한 좌석 수를 넘어선 경우
if (selectedCount >= currentCount) {
$(p.dot-list).text('좌석 예약이 완료되었습니다.');
$('#document-modal').css('display', 'flex');
} else {
$(this).addClass('selected');
selectedCount++; // selectedCount--는 현재 선택된 좌석의 수를 1 증가
selectedSeats.push($(this).data('seatno')); // selectedSeats라는 배열에 현재 선택한 좌석의 번호를
// 선택된 좌석 번호를 화면에 표시
$('.seat-position').eq(selectedCount-1).text(seatno).css('background-color', 'blue');
$(this).css('background-color', 'black');
$(this).css('color', '#c9b459');
}
```

4-5. 예매 및 좌석 선택

```
// 초기화 버튼
function resetSeats() {
    // 좌석 초기화
    $('.selected').removeClass('selected').prop('disabled', false).css('background-color', 'white').css('color', 'black');
    selectedSeats = [];
    selectedCount = 0;
    // 인원 수 초기화
    $('.number .now').text('0');
    $('.money em').text('0'); // 가격을 0으로 초기화
    // getMaxSeat() 함수는 좌석의 최대 개수를 반환하는 함수입니다. availableSeats 변수에 getMaxSeat()의 반환값을 할당함
    availableSeats = getMaxSeat();
    // 연령대 선택 초기화
    $('.check-people .number .now').text('0');
    currentCount = 0;

    // 인원 수 업데이트
    peopleCount();
    // 선택좌석
    $('#div.seat-position').text('-');

}
// 초기화 버튼
$('.button-reset').click(resetSeats);
```

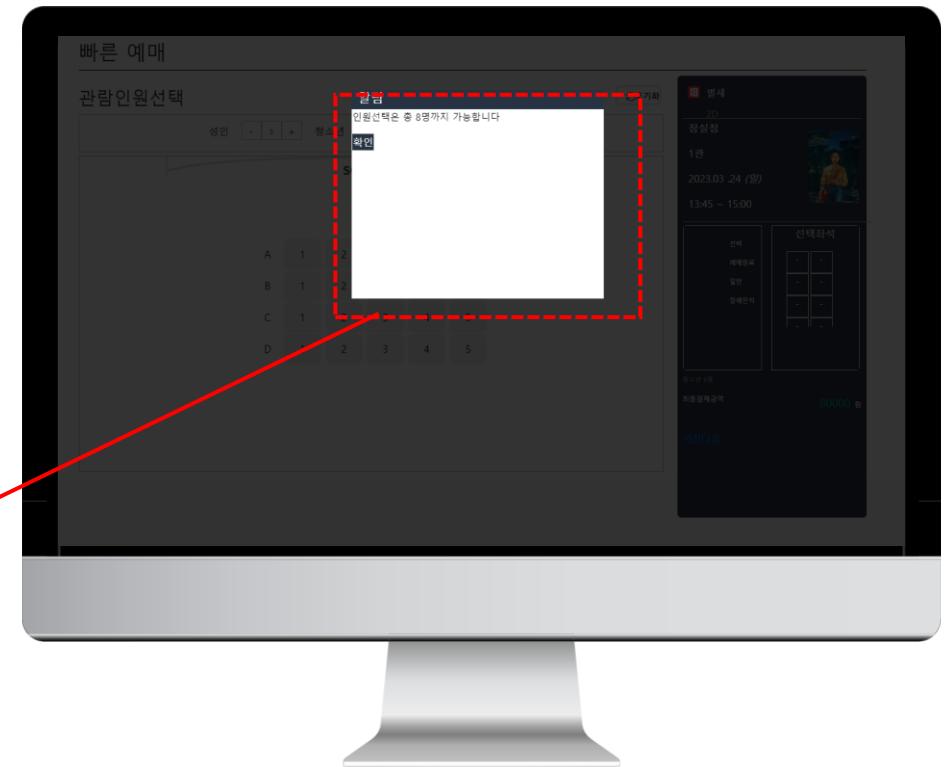
```
$('.dot-list').text("인원선택은 총 8명까지 가능합니다");
$('#document-modal').css('display', 'flex');
```

```
$('.dot-list').text('좌석 예약이 완료되었습니다.');
$('#document-modal').css('display', 'flex');
```

```
$('.btn-document-modal').click(function(){
    $('#document-modal').css('display', 'none');
})
```

설명 Alert 대신 Document – model 구현

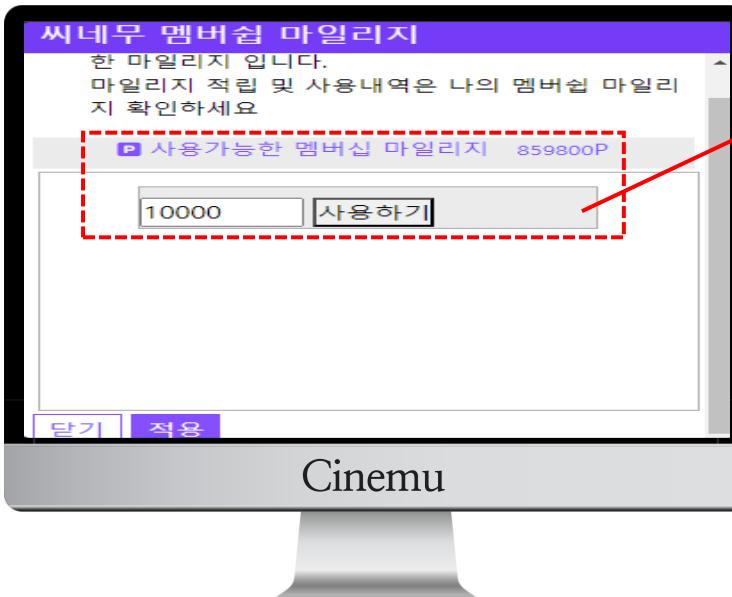
좌석 선택 – 초기화 & 팝업 창



영화 결제 – 마일리지 차감



Cinemu



Cinemu

```
<!-- 결제 마일리지 사용시 마일리지 차감 update -->
<update id="MileageDed" parameterType="map">
    UPDATE login SET lmileage = lmileage - #{subtractAmount,jdbcType=INTEGER}
    WHERE lid = #{lid,jdbcType=VARCHAR}
</update>
```

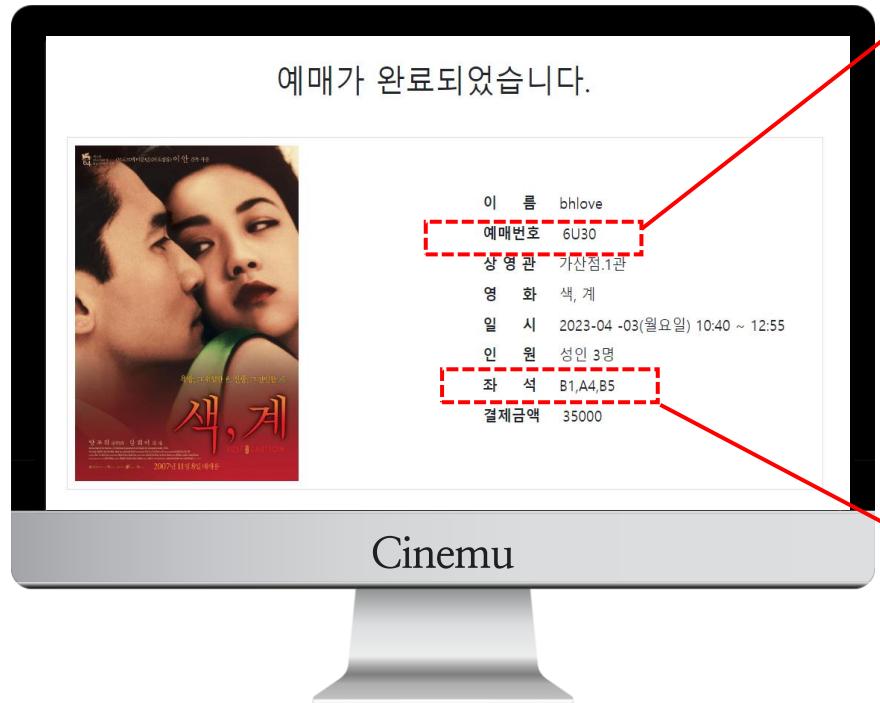
```
$ajax({
    type: "POST",
    url: "${pageContext.request.contextPath}/Mileage/MileageDed",
    dataType: 'json',
    data: {
        lid: uid,
        subtractAmount: inputPoint
    },
    success: function(data) {
        $('#p.discount-pay').text(inputPoint);
        let currentPay = parseInt('${ticket.totalPrice}');
        let minus = parseInt($('#p.discount-pay').text());
        let numberTotal = currentPay * 1;
        let currentTotal = numberTotal - minus;
        if (currentTotal >= 0) {
            $('#p.final-total').text(currentTotal);
        } else {
            $('#p.final-total').text(currentPay);
        }
        let point = parseInt(inputPoint);
        if (point >= 0) {
            $('#p.point-discount').text(point);
        } else {
            $('#p.point-discount').text(0);
        }
        $('#point-modal').css('display', 'none');
    },
    error: function(e) {
        console.log(e);
        alert('마일리지 차감에 실패했습니다. 다시 시도해주세요.');
    }
},
```

설명

결제 시 사용자의 마일리지 차감 기능 구현
 $lmilage = lmilage - \{subtractAmount\}$ 사용하여
 사용 가능한 마일리지에서 숫자 입력시 차감 되어 결제
 시 사용할 수 있음.

4-5. 예매 및 좌석 선택

영화 결제 – 결제 완료



Cinemu

설명

결제 완료 시 좌석 booked=1로 들어갈 수 있게 기능 구현
좌석 2개 이상 선택을 위해 Dynamic SQL 사용
사용자 중복 없이 좌석 선택 가능

```
function generateBookingNo() {
    let bookingNo = '';
    const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'; // 사용할 문자들
    const length = 5; // 예매번호 길이
    for (let i = 1; i < length; i++) {
        const randomIndex = Math.floor(Math.random() * characters.length); // 무작위로 인덱스를 선택
        bookingNo += characters[randomIndex]; // 선택된 문자를 예매번호에 추가
    }
    return bookingNo;
}
const bookingNo = generateBookingNo(); // 예매번호 생성
```

설명 예매번호 랜덤 생성

```
<update id="updateBooked" parameterType="java.util.List">
    UPDATE seat
    SET booked = 1
    WHERE booked = 0 and seatno IN
        <foreach collection="List" item="seatno" open="(" separator="," close="")">
            #{seatno}
        </foreach>
        and theater_id= #{theater_id} and screening_id = #{screening_id}
</update>
```



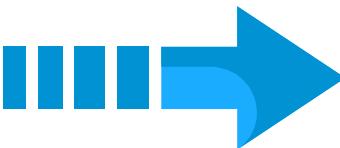
SEATNO	SEATNO	MBYEOL	MBHANG	BOOKED
1 A1	A	1		0
2 A2	A	2		0
3 A3	A	3		0
4 A4	A	4		1
5 A5	A	5		0
6 B1	B	1		1
7 B2	B	2		0
8 B3	B	3		0
9 B4	B	4		0
10 B5	B	5		1

4-6. 스토어 및 결제

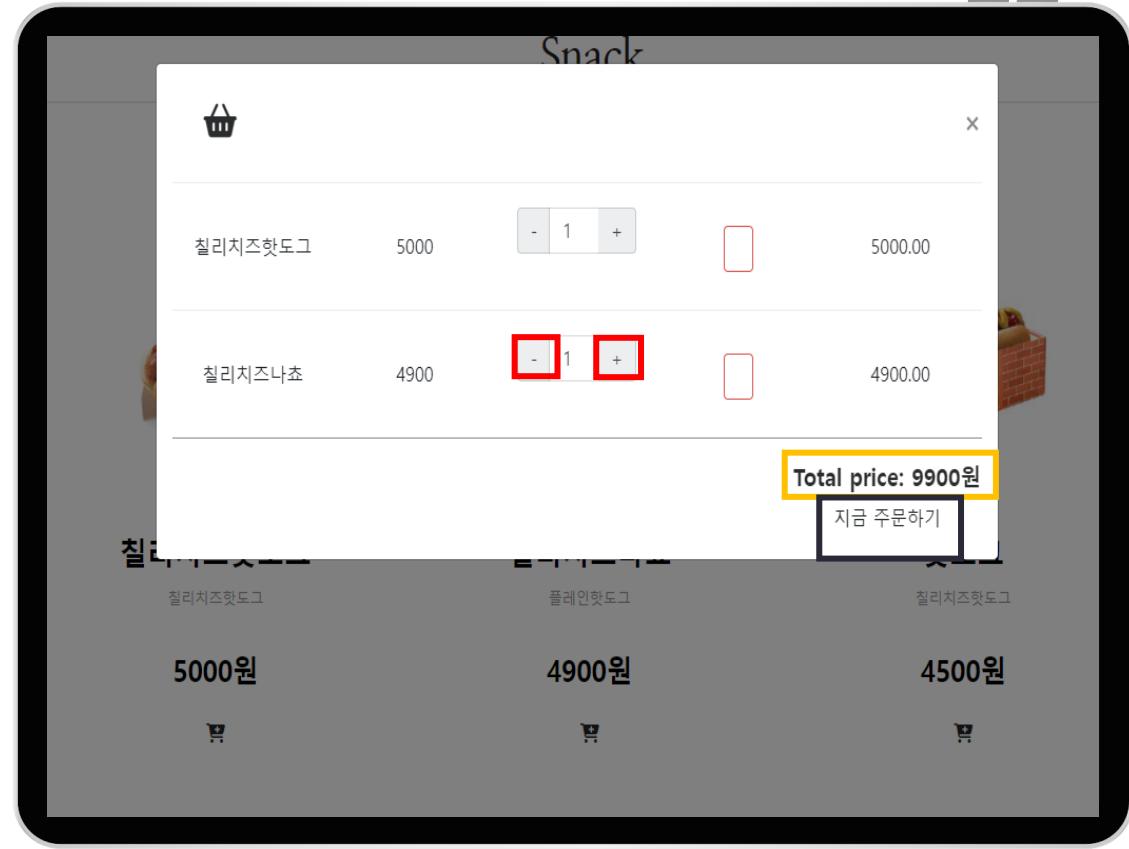
스낵 리스트



Add to cart
Click



장바구니



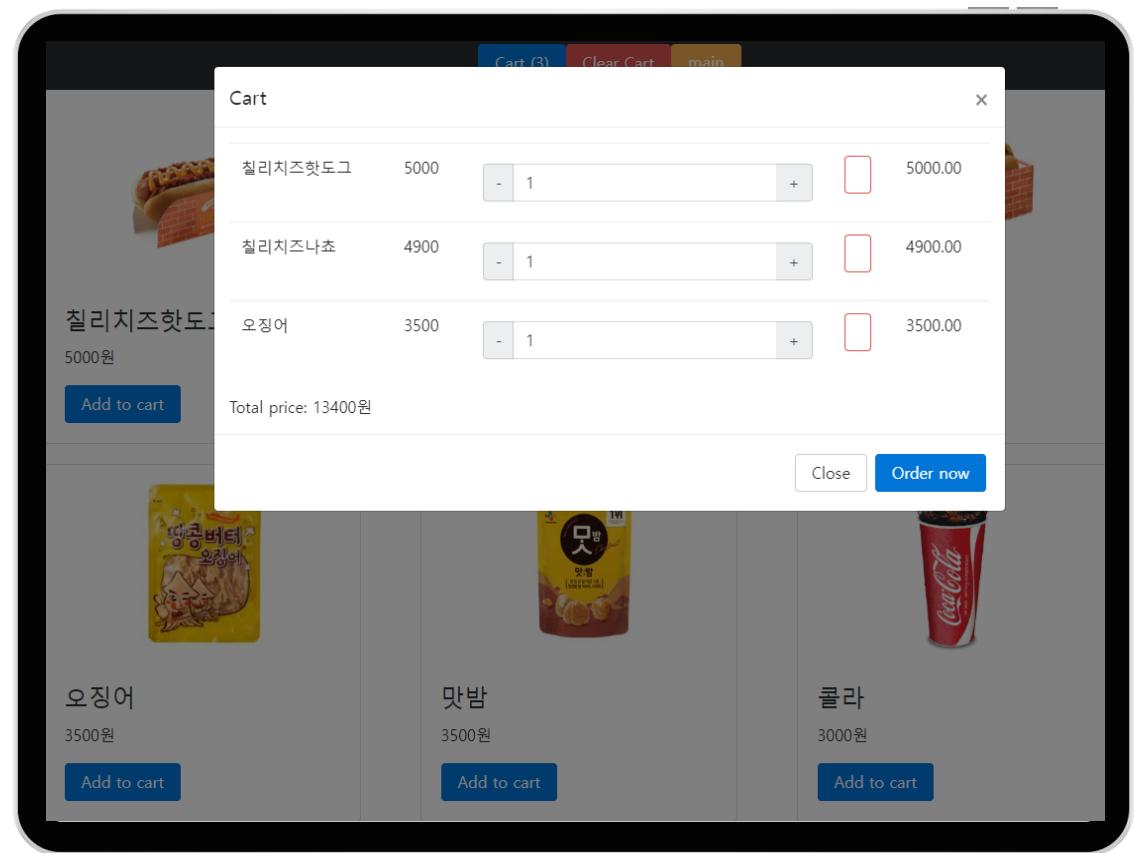
01 합계금액

02 각 스낵 별 개수 더하기, 빼기

03 결제

4-6. 스토어 및 결제

장바구니



```
<insert id="adddbuy" parameterType="java.util.List">
  <!-- for(String e : list) -->
  <foreach collection="list" item="e" separator=" "
    open="insert all" close="select * from dual">
    into snackbuy
    values(snackbuy_seq.nextVal,#{e.lid, jdbcType=VARCHAR},
      #{e.poppay, jdbcType=INTEGER},#{e.popname, jdbcType=VARCHAR},#{e.snacksaram, jdbcType=INTEGER})
  </foreach>
</insert>
```

설명

Insert all을 사용하여 장바구니 리스트에 있는 모든 품목 결제

JavaScript



```
function displayCart() {
  let trying = '';
  let pricea = '';
  let item = '';
  let member = {};
  let cartArray = shoppingCart.listCart();
  let output = '';
  let inputval = '';
  for(var i in cartArray) {
    //Cart 리스트 생성
    output += "<tr>" +
      "<td class='trying'>" + cartArray[i].name + "</td>" +
      "<td class='pricea'>" + cartArray[i].price + "</td>" +
      "<td><div class='input-group-addon btn btn-primary' data-name=" + cartArray[i].name + ">" +
      "<input type='number' class='item-count form-control' data-name='" + cartArray[i].name + "' value='" + cartArray[i].count + "'>" +
      "<button class='plus-item btn btn-primary input-group-addon' data-name=" + cartArray[i].name + ">+</button></div></td>" +
      "<td><button class='delete-item btn btn-danger' data-name=" + cartArray[i].name + ">X</button></td>" +
      "<td>" +
      "<td>" + cartArray[i].total + "</td>" +
      "</tr>";
    inputval += '<input type="hidden" name="poppay" class="pricea" value="">' +
      '<input type="hidden" name="popname" class="trying" value="">' +
      '<input type="hidden" name="snacksaram" class="snacksaram" value="">';
    /*
    inputval += '<input type="hidden" name="poppay" class="pricea" value="">' +
      '<input type="hidden" name="popname" class="trying" value="">' +
      '<input type="hidden" name="snacksaram" class="snacksaram" value="">';
    */
    // input value 생성
  }
  $('#paybtn').html(inputval);
  $('.show-cart').html(output);
  $('.total-cart').html(shoppingCart.totalCart());
  $('.total-count').html(shoppingCart.totalCount());
}
```

설명

Add to cart, +버튼 클릭 시 파라미터 추가

Controller

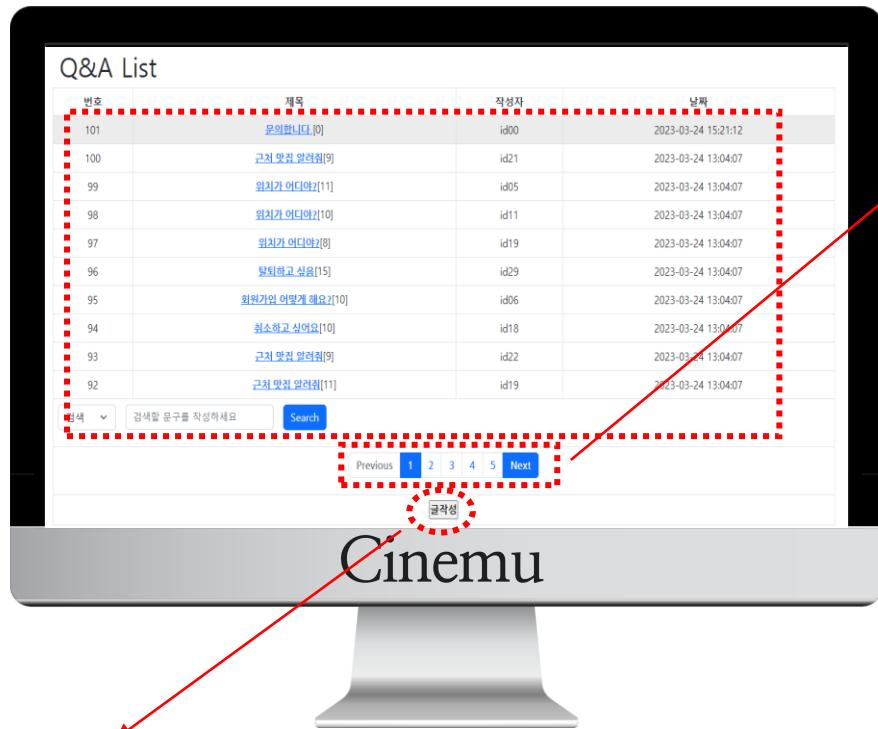
```
@RequestMapping(value = "/snackbuy")
public String addSnack(String[] poppay, String[] popname, String[] snacksaram, HttpSession session) {
  String lid = (String) session.getAttribute("sessionID");
  List<SnackBuyDTO> svolist = new ArrayList<SnackBuyDTO>();
  for(int i=0; i<poppay.length; i++) {
    SnackBuyDTO svo = new SnackBuyDTO();
    svo.setLid(lid);
    svo.setPoppay(Integer.parseInt(poppay[i]));
    svo.setPopname(popname[i]);
    svo.setSnacksaram(Integer.parseInt(snacksaram[i]));
    svolist.add(svo);
  }
  snackBuyDAOInter.snackBuy(svolist);
  return "redirect:/main";
}
```

설명

Form에서 전달 한 파라미터 배열로 받기

4-7. QnA 게시판

Q&A 게시판 페이지



controller

- 페이지 당 10개의 게시글
- 페이지당 5개의 페이징 블록 설정

```
private int numPerPage = 10; // 한 페이지당 보여질 게시물 수
private int pagePerBlock = 5; // 한 블럭당 보여질 페이지 수

@RequestMapping(value = "/qalist")
public String listBoard(Model model, String cPage, String searchType, String searchValue) {

    PageSearchDTO vo = new PageSearchDTO();
    vo.setSearchType(searchType);
    vo.setSearchValue(searchValue);
    totalRecord = qnABoardDAOInter.getCnt(vo);
    totalPage = (int) Math.ceil(totalRecord / (double) numPerPage);
    totalBlock = (int) Math.ceil((double) totalPage / pagePerBlock);
    String s_page = cPage;
    if (s_page != null) {
        nowPage = Integer.parseInt(s_page);
    }
    beginPerPage = (nowPage - 1) * numPerPage + 1;
    endPerPage = (beginPerPage - 1) + numPerPage;
    vo.setBegin(beginPerPage);
    vo.setEnd(endPerPage);
    List<QnABoardDTO> list = qnABoardDAOInter.getSearchList(vo);
    int startPage = (int) ((nowPage - 1) / pagePerBlock) * pagePerBlock + 1;
    int endPage = startPage + pagePerBlock - 1;
    if (endPage > totalPage) {
        endPage = totalPage;
    }
    model.addAttribute("list", list);
    model.addAttribute("nowPage", nowPage);
    model.addAttribute("startPage", startPage);
    model.addAttribute("endPage", endPage);
    model.addAttribute("pagePerBlock", pagePerBlock); // 설정된 5
    model.addAttribute("totalPage", totalPage);
    model.addAttribute("searchType", vo.getSearchType());
    model.addAttribute("searchValue", vo.getSearchValue());
    model.addAttribute("totalRecord", totalRecord);
    return "qna/qnaList";
}
```

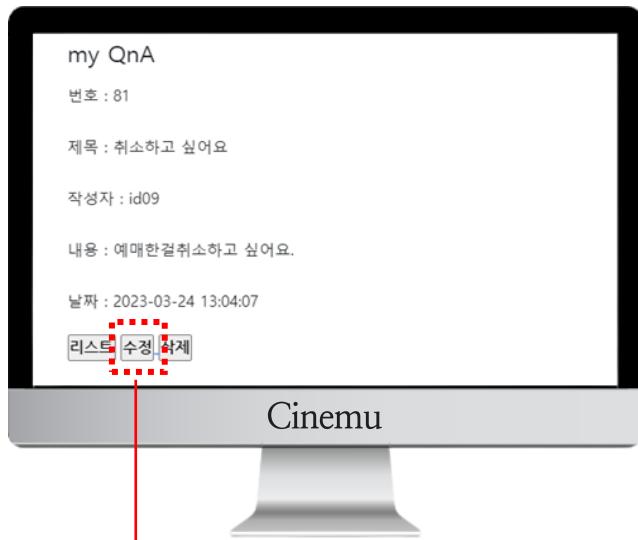
jsp

session 값을 받아서, 로그인 시에만 글작성 버튼이 노출

```
<tr>
    <c:choose>
        <c:when test="${sessionScope.sessionID != null }">
            <td colspan=4 style="text-align: center"><a href="#">\\${pageContext.request.contextPath}/qaboard/qafarm" type="button" value="글작성" id="wbtn" class="button-l">></a>
```

4-7. QnA 게시판

Q&A 게시판 상세페이지



controller

- 게시판 상세페이지 및 수정 / 삭제 컨트롤러
- 문의글 삭제 시 게시판 리스트로 이동
- 수정 버튼 클릭 시 수정 form 이동
- 수정 완료 후 상세 페이지 이동

```
@GetMapping(value = "/qadetail")
public ModelAndView detail(int no) {
    ModelAndView mav = new ModelAndView("qna/qnaDetail");
    System.out.println(no);
    QnABoardDTO vo = qnABoardDAOInter.detailBoard(no);
    List<QnACommDTO> listcomm = qnABoardDAOInter.listQAComm(no);
    mav.addObject("listcomm", listcomm);
    System.out.println(vo);
    mav.addObject("vo", vo);
    return mav;
}

@RequestMapping(value = "/qaupdate")
public String updateForm(QnABoardDTO vo) {
    qnABoardDAOInter.upBoard(vo);
    return "redirect:/qaboard/qadetail?no="+vo.getNo();
}

@GetMapping(value = "/qadelete")
public String deleteBoard(int no) {
    qnABoardDAOInter.delete(no);
    return "redirect:/qaboard/qalist";
}
```

jsp

session 값이 작성자와 일치 할 경우에만, 수정 / 삭제 버튼이 노출

```
<a href="qalist"> <input
    type="button" value="리스트" id="ListBtn" class="button-l" /></a>
<c:choose>
<c:when test="${sessionScope.sessionID == vo.writer }">
    <a href="qaupform?no=${vo.no }">
        <input type="button" value="수정" id="wbtn" class="button-l" />
    </a>
    <a href="qadelete?no=${vo.no }"> <input type="button" value="삭제"
        id="delBtn" class="button-l" /></a>
</c:when>
</c:choose>
```

Q&A 게시판 댓글

설명

관리자(admin)이 문의글에 댓글을 달았을 경우, 문의글 작성자의 E-Mail로 해당 내용 전송

```

<select id="linkedmail" parameterType="int" resultMap="qcnLog">
    select
        l.lemail, l.lname, a.content, a.subject from login l, (
            select q.content,q.writer,q.mdate, q.subject from qaboard q where
            q.no=#{no}) a where l.lid = a.writer
</select>
<resultMap type="mvo" id="qcnLog">
    <result property="lemail" column="lemail" />
    <result property="lname" column="lname" />
    <collection property="qnaboard" javaType="java.util.List"
        resultMap="qcnemail" ofType="qvo" />
</resultMap>
<resultMap type="qvo" id="qcnemail">
    <result property="content" column="content" />
    <result property="subject" column="subject" />
</resultMap>

@RequestMapping(value = "/qcomminsert")
public ModelAndView addQaComm(QnACommDTO qcvo) {
    qnABoardDAOInter.addQAComm(qcvo);
    int no = qcvo.getMcode();
    ModelAndView mav = new ModelAndView("redirect:/qaboard/qadetail?no=" + no);
    String id = qcvo.getMwriter();
    if (id.equals("admin")) {
        LoginDTO lvo = qnABoardDAOInter.linkdEmail(no);
        List<QnABoardDTO> qlist = lvo.getQnaboard();
        String sujetc = qlist.get(0).getSubject();
        String content = qlist.get(0).getContent();
        String emailFromRecipient = "*****@naver.com";
        String emailSubject = lvo.getLname() + "님의 문의에 대한 담당자의 답변이 등록되었습니다.";
        String bar = "\n-----\n";
        String aa = "\n\n\n 문의하신 내용 " + bar + "제목 :\t" + sujetc + "\n" + content + "\n\n\n";
        String emailMessage = "답변 \n" + qcvo.getMcontent() + aa;
        String emailToRecipient = lvo.getLeemail();

        mailSender.send(new MimeMessagePreparator() {
            public void prepare(MimeMessage mimeMessage) throws Exception {
                MimeMessageHelper mimeMsgHelperObj = new MimeMessageHelper(mimeMessage, true, "UTF-8");
                mimeMsgHelperObj.setTo(emailToRecipient);
                mimeMsgHelperObj.setFrom(emailFromRecipient);
                mimeMsgHelperObj.setText(emailMessage);
                mimeMsgHelperObj.setSubject(emailSubject);
            }
        });
    }
    return mav;
}

```

4-8. 데이터 수집

데이터 수집 – Beautiful Soup / Naver 스크래핑



```
idxl, titlel, contl, pointl = [], [], [], []
userl, datel, mno = [], [], []

def test(num):
    for u in range(num):
        url = 'https://movie.naver.com/movie/point/af/list.naver?page={}'.format(u+1)
        response = requests.get(url)
        soup = BeautifulSoup(response.text, 'html.parser')

        # 번호
        idx = soup.find_all('td', class_='ac num')
        # 제목
        title = soup.find_all('a', class_='movie color_b')
        # 평점
        cont = soup.find_all('td', class_='title')
        # 평점
        point = soup.find_all('div', class_='list_netizen_score')
        # 날짜
        daten = soup.select('tbody td:nth-child(3)')
        # ID
        user = soup.find_all('a', class_='author')
        # 영화 번호
        mnoList = soup.find_all('td', class_='title')

        for u in range(len(idx)):
            idxl.append(idx[u].text)
            titlel.append(title[u].text)
            contl.append(cont[u].br.next_sibling.replace('\n', '').replace('#t', '')) 
            pointl.append(point[u].em.text)
            datel.append(daten[u].br.next_sibling)
            userl.append(user[u].text)
            mno.append(mnoList[u].a.attrs['href'].split('=')[2].split('&')[0])

    print('...')

df_movie = pd.DataFrame([idxl, mno, contl, pointl, userl, datel, titlel]).T
return df_movie

myMovie.columns=['MIDX', 'MNO', 'MCCONTENT', 'MPOINT', 'MUSER', 'MDATE', 'MTITLE']
#myMovie.columns=['MIDX', 'MTITLE', 'MCCONTENT', 'MPOINT', 'MUSER', 'MDATE']

myMovie.to_csv('movieComm_Naver.csv')
```

네이버 영화 리뷰 사이트에서 리뷰데이터를 뷰티풀숩을 이용하여 스크래핑

수집한 데이터를 Pandas의 DataFrame 객체로 변환후, CSV로 저장



Cinemu

MIDX	MNO	MCCONTENT	MPOINT	MUSER	MDATE	MTITLE
0	18817706	11348	항상 볼까말까 고민했었는데 꽤 경찮았어요 물의 로고 깨끗하고 친근 Think...	8	num****	23.03.22 이미데이션 개임
1	18817705	198413		10	emil****	23.03.22 헤어질 결심
2	18817704	215932	일도적인 영상미와 연출력에도 놓장 인물들의 감작스러운 감정선을 따라가기 어려웠습니다...	6	gkrd****	23.03.22 스즈메의 문단속
3	18817703	72072	티비에서 하면서 다시 봤는데 마지막 장면에 남편이 표송하면서 '사랑해'라고 말하는데...	8	kill****	23.03.22 라스트 낙인
4	18817702	215932	내용률이 조금 부실하지만 눈이 되게 만족스럽다	8	bjh0****	23.03.22 스즈메의 문단속
...
95	18817701	191430	원작이 너무 재밌었을 것	4	18_y****	23.03.22 자발
96	18817700	220468	케이트, 대체 당신 못하는 게 뭐요?	7	palm****	23.03.22 TAR 탄트
97	18817699	187923	꿈과 현실을 공유하는게 진정한 사랑이지	7	bloo****	23.03.22 소울메이트
98	18817698	198413	주위에서 다른 인생영화라고 극찬해서 너무 기대하고 보다가 저작하게 나락간비로 이 영...	1	6166****	23.03.22 헤어질 결심
99	18817697	208530	처음봤을때는 참 힘들었는데.. 연기들도 어색하고.. 두번째보면 더 잘있는 영화....	10	itgi****	23.03.22 강풀

Cinemu

4-8. 데이터 수집

데이터 수집 – Selenium / Daum 크롤링



```
for i in range(len(movieNum)):
    num = movieNum[i].attrs['href'].split('=')[1]
    urlCom = f"https://movie.daum.net/moviedb/grade?movieId={num}"
    print(urlCom)
    driver = webdriver.Chrome(executable_path=os.path.abspath("chromedriver"))
    driver.get(urlCom)
    time.sleep(3)
    try:
        element = driver.find_element(By.CLASS_NAME, 'list_comment')
        title = driver.find_element(By.CLASS_NAME, 'tit_movie').find_element(By.CLASS_NAME, 'txt_tit').text
        print(title)

        while True:
            btn = element.find_element(By.XPATH, '//*[@id="alex-area"]/div/div/div[3]/div[1]/button')
            btn.click()
            time.sleep(0.5)
    except:
        print('end')
        pass

    point = element.find_elements(By.CLASS_NAME, 'ratings')
    comm = element.find_elements(By.CLASS_NAME, 'desc_txt')
    date = element.find_elements(By.CLASS_NAME, 'txt_date')
    user = element.find_elements(By.CLASS_NAME, 'info_author')
    print(len(comm))
    print(title)

    for e in range(len(comm)):
        idkl.append(f'{num}{"\t"}{title}')
        titl.append(title)
        pointl.append(point[e].text)
        contl.append(comm[e].text)
        datel.append(date[e].text)
        userl.append(user[e].find_element(By.TAG_NAME, 'a').text)

    driver.close()
except:
    driver.close()

df_movie = pd.DataFrame([idkl, titl, contl, pointl, userl, datel]).T
df_movie.columns=['MIDX', 'MTITLE', 'MCONTENT', 'MPOINT', 'MUSER', 'MDATE']
df_movie

df_movie.to_csv('work/moviedata2.csv')
```

다음의 영화 리뷰사이트에서 셀레니움을 활용하여 데이터를 수집



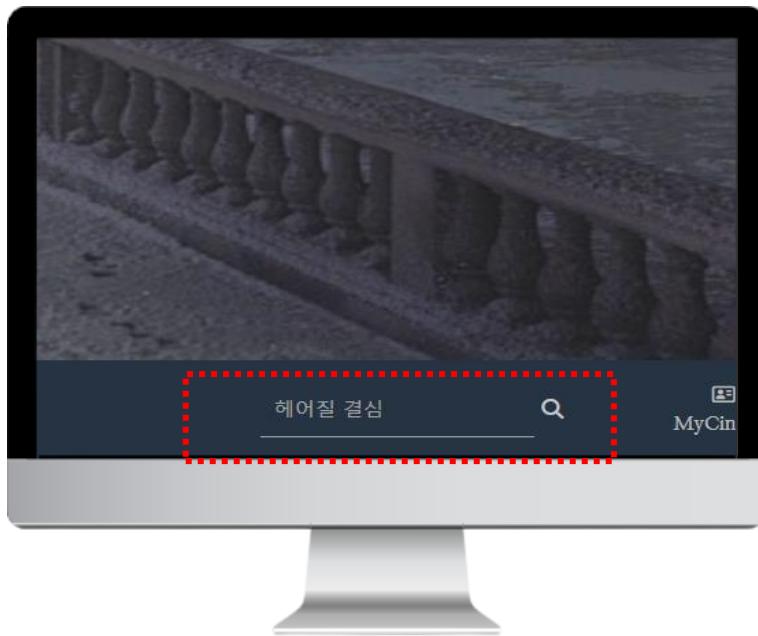
Cinemu

MIDX	MNO	MCONTENT	MPOINT	MUSER	MDATE	MTITLE
0	18817706	11348	항상 볼까말까 고민했었는데 꽤 걸쳤었어요 그들의 로고 깨끗하고 친근 Think...	8	num****	23.03.22 이미지아선 개임
1	18817705	198413		10	emil***	23.03.22 헤어질 결심
2	18817704	215932	일도적인 영상미와 연출력에도 놓장 인물들의 감작스러운 감정선을 따라가긴 어려웠습니다...	6	gkrd***	23.03.22 스즈메의 문단속
3	18817703	72072	티비에서 하면서 다시 봤는데 마지막 장면에 남편이 표송하면서 '사랑해'라고 말하는데...	8	kill***	23.03.22 라스트 낙인
4	18817702	215932	내용률이 조금 부실하지만 눈이 되게 만족스럽다	8	bjh0***	23.03.22 스즈메의 문단속
...
95	18817701	191430	원작이 너무 재밌었을 것	4	18_y***	23.03.22 자발
96	18817700	220468	케이트, 대체 당신 못하는게 뭐요?	7	palm***	23.03.22 TAR 탄트
97	18817709	187923	꿈과 현실을 공유하는게 진정한 사랑이지	7	bloo***	23.03.22 소울메이트
98	18817708	198413	주위에서 다른 인생영화라고 극찬해서 너무 기대하고 보다가 저작하게 나락간비로 이 영...	1	6166***	23.03.22 헤어질 결심
99	18817709	208530	처음봤을때는 참 힘들었는데.. 연기들도 어색하고.. 두번째보면 더 잘있는 영화....	10	itgi***	23.03.22 강풀

Cinemu

수집한 데이터를 Pandas의 DataFrame 객체로 변환후, CSV로 저장

전체 검색 - 검색



mapper

- 검색 키워드를 입력 시, 검색 결과의 항목별 검색의 숫자를 출력하며, 그 키워드를 붉은색으로 강조하여 출력
- 검색 키워드를 빈 칸으로 두고 검색하였을 때, rownum을 설정하여 데이터 베이스에 저장되어있는 영화 중 무작위로 1 건 출력

```
<!-- 영화 검색 -->
<select id="movie" parameterType="String" resultType="muvo">
    select mno,mimg,
    <![CDATA[replace(mname,#{searchValue},
    '<span style="color:red">${searchValue}</span>')mname]]>
    from movieup where
    mname like '%'||#{searchValue}|| '%' or
    mdirec like '%'||#{searchValue}|| '%' or
    mstory like '%'||#{searchValue}|| '%' or
    mactor like '%'||#{searchValue}|| '%'
</select>

<!-- 랜덤 영화이름 출력 -->
<select id="random" resultType="String">
SELECT mname FROM (
SELECT mname FROM movieup ORDER BY DBMS_RANDOM.VALUE) WHERE ROWNUM = 1
</select>
```

전체 검색 – 검색 결과 및 더보기 (1)



jsp

- 가장 최근에 입력된 3개의 데이터가 출력되며, 검색된 데이터가 없을 때는 항목의 탭과 [더보기] 버튼이 비활성화
- 더 많은 데이터를 보고 싶을 때는 [더 보기]버튼이나 해당 항목의 탭을 클릭하면 해당 항목탭으로 이동

```

<ul class="nav nav-tabs" id="myTab" role="tablist">
    <li class="nav-item" role="presentation">
        <button class="nav-link active" id="totalSearch-tab"
            data-bs-toggle="tab" data-bs-target="#totalSearch-tab-pane"
            type="button" role="tab" aria-controls="totalSearch-tab-pane"
            aria-selected="true">전체</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="nav-link" id="movlist-tab" data-bs-toggle="tab"
            data-bs-target="#movlist-tab-pane" type="button" role="tab"
            aria-controls="movlist-tab-pane" aria-selected="false">전체영화
            (${mcnt })</button>
    </li>
</ul>

<label>영화 검색결과</label>
<button type="button" id="moremov">더보기</button>
<col>
<c:forEach var="e" items="${mlist}" begin="0" end="3">
    <li><div style="float: left; width: 25%;">
        <div style="width: 90%; height: 90%">
            
        </div>
        <div>
            <a
                href="${pageContext.request.contextPath}/movies/moviedetail?no=${e.mno }" ${e.mname}>
            </a>
        </div>
    </li></c:forEach>
</c:forEach>

$(function() {
    if (${mcnt } === 0) {
        $("#movlist-tab").attr("disabled", true);
        $("#moremov").attr("disabled", true);
    }
    $('#moremov').click(function() {
        $("#movlist-tab").trigger("click");
    });
});

```

전체 검색 – 검색 결과 및 더보기 (2)

RestController

- 페이지에서 검색어(searchValue), 현재 페이지 번호(nowPage), 템 분류(code)의 값을 매개변수로 받음
- cPage 파라미터를 이용하여 현재 페이지 번호(nowPage)를 설정하고, numPerPage 파라미터를 이용하여 한 페이지당 보여질 게시물 수(numPerPage)를 설정
- 매개변수인 code 값에 따라 검색 대상이 다르며, 각 검색 결과를 ObjectMapper 클래스를 이용하여 JSON 형식으로 변환하여 반환

```

private int nowPage = 1; // 현재 페이지 값 -> 메뉴 페이지와 연동되는 변수
private int numPerPage = 10; // 한 페이지당 보여질 게시물 수
private int beginPerPage; // 각 페이지별 시작 게시물의 index값
private int endPerPage; // 각 페이지별 마지막 게시물의 index값

@RequestMapping(value = "/searchAjax", produces = "application/json;charset=utf-8")
public String searchAjax(String code, String searchValue, String cPage) {
    PageSearchDTO psvo = new PageSearchDTO();
    psvo.setSearchValue(searchValue);
    String s_page = cPage;
    if (s_page != null) {
        nowPage = Integer.parseInt(s_page);
    }
    beginPerPage = nowPage;
    endPerPage = beginPerPage+numPerPage-1;
    psvo.setBegin(beginPerPage);
    psvo.setEnd(endPerPage);
    String res = "";
    ObjectMapper mapper = new ObjectMapper();
    if(code.contains("QnA")) {
        List<QnABoardDTO> qlist = searchQueryInter.getQnaP(psvo);
        try {
            res = mapper.writeValueAsString(qlist);
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    } else if (code.contains("Review")) {
        List<MovieCommDTO> mclist = searchQueryInter.getMCommP(psvo);
        try {
            res = mapper.writeValueAsString(mclist);
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    } else if (code.contains("Mov")) {
        List<MovieUpDTO> movlist = searchQueryInter.getMovP(psvo);
        try {
            res = mapper.writeValueAsString(movlist);
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    }
    return res;
}

```

전체 검색 – 검색 결과 및 더보기 (3)

Ajax 및 Json Data

- 각 항목의 탭에는 최초 10개의 데이터가 출력되며, [More]버튼을 누를 경우 10개씩 추가하여 출력
- 더 이상의 출력될 데이터가 없을 경우에 [More]버튼을 숨김



```

$( '#movcomm-tab' ).click( function() {
  let ajaxUrl = ".../searchAjax?code=Review&searchValue=" + sv + "&cPage=";
  let numPerPage = $('#mcbody > tr').length;
  let obj = $(this).attr('aria-controls');
  // Ajax 시작
  ajaxList(ajaxUrl, numPerPage, obj, function(data) {
    console.log("data.length==>" + data.length);
    let html = '';
    for (let i = 0; i < data.length; i++) {
      html += '<tr><td>' + data[i].commno + '</td>';
      html += '<td><a href="#">' + ${pageContext.request.contextPath}/movies/moviedetail?no='+data[i].mno+'</a>';
      html += data[i].comm + '</td>';
      html += '<td>' + data[i].mgood + '</td>';
      html += '<td>' + data[i].commdate + '</td></tr>';
    }
    $('#mcbody').append(html);
  });
});

$('#mcbtn').click(function() {
  $("#movcomm-tab").trigger("click");
});

function ajaxList(ajaxUrl, numPerPage, obj, callback) {
  let currentPage = 1;
  const tableViewId = $('#' + obj).attr('id');
  const tableViewBody = $('#' + tableViewId + ' tbody');
  const moreButton = $('#' + tableViewId + ' button');
  currentPage += numPerPage;
  console.log("currentPage=====>" + currentPage);
  moreButton.text('Loading...');
  moreButton.attr('disabled', true);
  $.ajaxSetup({ cache: false });

  $.ajax({
    url: ajaxUrl + currentPage,
    success: function (data) {
      moreButton.text('More');
      moreButton.attr('disabled', false);
      callback(data);
      if (data.length < numPerPage) {
        moreButton.remove();
      }
    },
    error: function () {
      moreButton.text('Error');
    }
  });
}

```

Pagination.js

- 페이징 처리를 위해 jsonData를 가져오기 위한 Ajax Module 함수
- Callback으로 Ajax Data를 가져온 후 for문 사용하여 검색 결과 출력.

4-11. API (OpenWeatherMap)

Django



```
def weatherJP(request):
    api_key =
    city = "Seoul"
    url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&lang=en&units=metric"
    response = requests.get(url)
    weather_data = response.json()
    temp_celsius = round(weather_data['main']['temp'], 1) # 켈빈 -> 섭씨 변환 없이 단순히 소수점 1자리까지 반올림
    context = {
        'city': city,
        'temp': temp_celsius,
        'humidity': weather_data['main']['humidity'],
        'description': WEATHER_TRANSLATION[weather_data['weather'][0]['description']],
        'icon': f'{weather_data["weather"][0]["icon"]}.png',
        'STATIC_URL': '/static/',
    }
    # JSONP 방식으로 응답하기 위해 콜백 함수 이름을 가져옵니다.
    callback = request.GET.get('callback')
    # JsonResponse를 사용하여 JSON 형식으로 응답합니다.
    response_data = json.dumps(context)
    if callback:
        # 콜백 함수 이름을 JSON 데이터와 함께 감싸서 JSONP 형식으로 응답합니다.
        response_data = f'{callback}({response_data})'
    return HttpResponse(response_data, content_type="application/javascript")
else:
```

JSONP

Ajax



```
// 날씨 받은거 코드
$.ajax({
    type: "GET",
    url: "http://192.168.0.189:9000/recomend/weatherJP",
    dataType: "jsonp",
    success: function(response) {
        // 성공적으로 응답을 받은 경우 처리할 코드
        console.log(response.city); // 예시: 'Seoul'
        console.log(response.temp); // 예시: 12.3
        console.log(response.humidity); // 예시: 45
        console.log(response.description); // 예시: 'clear sky'
        console.log(response.icon); // 예시: '01d.png'

        let tr = "";
        tr += "<p>오늘의 날씨</p>";
        tr += "<p>도시: " + response.city + "</p>";
        tr += "<p>온도: " + response.temp + "</p>";
        tr += "<p>습도: " + response.humidity + "</p>";
        tr += "<p><img src=${pageContext.request.contextPath}/resources/img/" + response.icon + "></p>";

        $('#moviewheather').html(tr);
    },
    error: function(xhr, status, error) {
        // 응답에 실패한 경우 처리할 코드
        console.log("Error: " + error);
    }
});
```

OpenWeatherMap API를 사용하여 서울의 현재 기상 상황을 가져옴

- API 요청 URL을 만들어서 requests 모듈을 사용하여 API에 요청합니다.
- API에서 받아온 응답을 json() 메서드를 사용하여 파이썬 딕셔너리 형태로 변환합니다.
- 요청이 JSONP 형식인 경우 콜백 함수 이름을 가져와서 JSONP 형식으로 응답합니다.
- 그렇지 않은 경우 JSON 형식으로 응답합니다.

jQuery의 Ajax 메소드를 사용하여, 서버에 HTTP GET 요청을 보내고, 응답을 JSONP 형식으로 받아와 처리

dataType 속성에 "jsonp"을 설정하면, JSONP 형식으로 데이터를 받아옴. 요청이 성공적으로 처리되면, success 함수가 실행됩니다. response 매개변수에는 서버에서 받아온 JSONP 데이터가 담겨있으며, 이 데이터를 이용하여 HTML 문자열을 생성하고, moviewheather 요소에 출력합니다.

추천 – 데이터 정규화

두 영화간의 상관관계를 분석하여 비슷한 취향을 가진 사용자들이 선호하는 영화를 추천하는 모델

데이터를 판다스(Pandas)라이브러리의 `read_csv()`함수를 이용하여 호출.
호출한 데이터의 사용자(id)와 평점(mgood), 영화이름(mttitle), 영화장르(mtypes) 데이터 추출.
각각의 영화의 평가점수(mgood)를 기반으로 영화관의 유사성을 계산하기 위해 피벗테이블을 matrix변수에 저장

MNNO	MNO	LID	MGOOD	COMMNO	COMM	COMMDATE	MTITLE
0	101	126	id09	4	이터널스랑 블랙팬서보단 나온거길아요	2023-03-18	엔트맨과 와스프: 퀸텀메니아
1	102	101	id22	5	재밌는데 생각보다 평점이 낮아요. 왜지? 퀼링타임용으로 딴인데. ㅎㅎ 옥외티 하나...	2023-03-18	스마트폰을 떨어뜨렸을 뿐인데 디외비
2	103	113	id19	1	무엇을 말하려는 건지도 모르겠고, 이야기가 딱 시원하게 끝나는 것도 아님. 애매하게...	2023-03-18	아바타: 물의 길
3	104	144	id04	4	동서양을 막론하고 빛줄이란. 2d로 봤는데 불만했고 3d로 또 볼 예정입니다 최고의 ...	2023-03-18	치히로 상
4	105	153	id17	5	두번 세번 보고싶은 울림이 있는 영화!!	2023-03-18	헤어질 결심
...
23805	23906	149	id00	3	기대이하였다 그냥그랬음	2023-03-18	TAR 타르
23806	23907	123	id02	3	강 나쁘지않았음ㅎㅎㅎ	2023-03-18	그냥 가볍게 보기좋은ㅎㅎㅎ
23807	23908	136	id02	4	재미있는 액션영화!	2023-03-18	이미테이션 게임
23808	23909	138	id04	4	돈주고보면 아까웠을...	2023-03-18	라스트 나이트
23809	23910	107	id02	2	스즈매의 문단속		

23810 rows × 7 columns

MNO	LID	MTITLE	MGOOD	MTYPES
0	126	id09	엔트맨과 와스프: 퀸텀메니아	4 [액션]
1	126	id09	엔트맨과 와스프: 퀸텀메니아	4 [액션]
matrix = data.pivot_table(index='LID',columns='MTITLE',values='MGOOD')				
#matrix.ixn = data.pivot_table(index='LID',columns=['MTITLE','MNO'],values=['MGOOD'])				
matrix.head(2)				
MTITLE 6번 칸 TAR 강릉 귀멸의 칼날: 상현집결, 그 대대 그대 그대 대외 더 퍼스트 똑똑 라스 트나잇 ... 에브리 성 에브리 이미 테이 자백 조커 치히 카운트 콜 텔 헌섬 헤어질 결 LID id00 NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN NaN NaN NaN NaN NaN NaN 3.666667 id01 NaN NaN NaN 1.666667 NaN 4.75 NaN 4.083333 NaN NaN ... NaN NaN NaN NaN NaN 3.333333 NaN NaN NaN 4.000000				

2 rows × 32 columns

추천 – 영화 추천(JupyterNoteBook)

```

2 rows × 32 columns

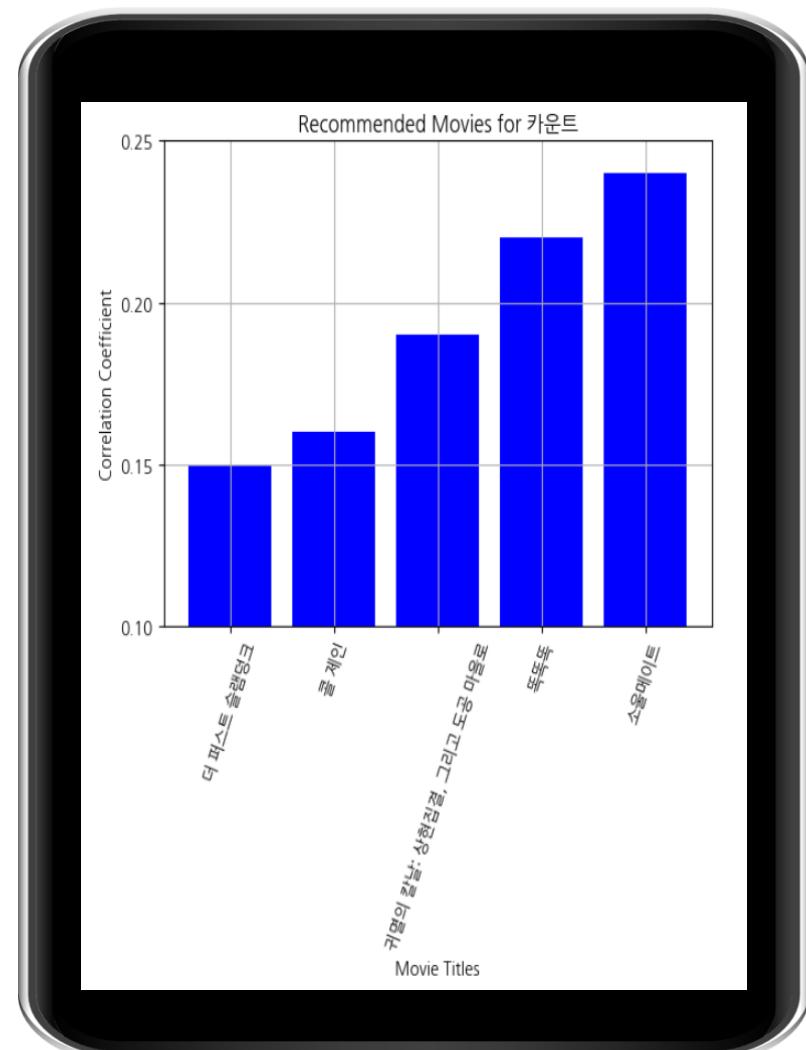
In [14]: GENRE_WEIGHT = 0.1
def pearsonR(s1,s2):
    s1_c=s1-s1.mean()
    s2_c=s2-s2.mean()
    return np.sum(s1_c*s2_c)/np.sqrt(np.sum(s1_c**2)*np.sum(s2_c**2))

In [15]: def recommend(input_movie, matrix, n, similar_genre=True):
    input_genres = data[data['MTITLE']==input_movie]['MTYPES'].iloc(0)[0]
    result = []
    for title in matrix.columns:
        if title == input_movie:
            continue
        cor = pearsonR(matrix[input_movie],matrix[title])
        # rating comparison
        if similar_genre and len(input_genres) >0:
            temp_genres = data[data['MTITLE']==title]['MTYPES'].iloc(0)[0]
            same_count = np.sum(np.isin(input_genres,temp_genres))
            cor += (GENRE_WEIGHT*same_count)
        if np.isnan(cor):
            continue
        else:
            result.append((title,'{:.2f}'.format(cor),temp_genres))
    result.sort(key=lambda r: r[1], reverse=True)
    return result[:n]

```

영화의 평점과 장르를 이용하여 관계를 계산하는 함수와, 가중치
 GENRE_WEIGHT, 비슷한 장르의 영화일수록 가중치를 더해 더 적합한 추천
 영화를 선택하는 함수 recommend를 코딩

댓글데이터에 포함된 영화이름을 입력하여, 이 영화와 유사한 영화 리스트를
 출력하여 시각화



4-12. 추천시스템

추천 -영화 추천(pyCharm-Django)

```
def get_recommendationsJP(request):
    pathv = '/home/kosmo/python/workspace/myweb/mvrd/static/dataset/'
    gdata = pd.read_csv(pathv+'movUp.csv')[['MNO', 'MTYPES']]
    mdata = pd.read_csv(pathv+'movComm.csv')[['MNO', 'LID', 'MTITLE', 'MGOOD']]
    meta = mdata[['MNO', 'LID', 'MTITLE', 'MGOOD']]
    genre = gdata[['MNO', 'MTYPES']]
    meta.MNO = pd.to_numeric(meta.MNO, errors='coerce')
    genre.MNO = pd.to_numeric(genre.MNO, errors='coerce')
    genre['MTYPES'] = genre['MTYPES'].str.split(',')
    data = pd.merge(meta, genre, on='MNO', how='inner')
    matrix = data.pivot_table(index='LID', columns='MTITLE', values='MGOOD')

    GENRE_WEIGHT = 0.1

    def pearsonR(s1, s2): ←
        s1_c = s1 - s1.mean()
        s2_c = s2 - s2.mean()
        return np.sum(s1_c * s2_c) / np.sqrt(np.sum(s1_c ** 2) * np.sum(s2_c ** 2))

    def recommend(input_movie, matrix, n, similar_genre=True): ←
        input_genres = data[data['MTITLE'] == input_movie]['MTYPES'].iloc[0][0]
        result = []
        for title in matrix.columns:
            if title == input_movie:
                continue
            cor = pearsonR(matrix[input_movie], matrix[title])
            if similar_genre and len(input_genres) > 0:
                temp_genres = data[data['MTITLE'] == title]['MTYPES'].iloc[0][0]
                same_count = np.sum(np.isin(input_genres, temp_genres))
                cor += (GENRE_WEIGHT * same_count)
            if np.isnan(cor):
                continue
            else:
                result.append((title, '{:.2f}'.format(cor), temp_genres))
        result.sort(key=lambda r: r[1], reverse=True)
        return result[:n]
```

Cross 통신으로 windows의
spring에 Jsonp방식으로 보내
기 위한 코드

영화의 평점과 장르를 이용하
여 관계를 계산

비슷한 장르의 영화일수
록 가중치를 더해 더 적
합한 추천 영화를 선택

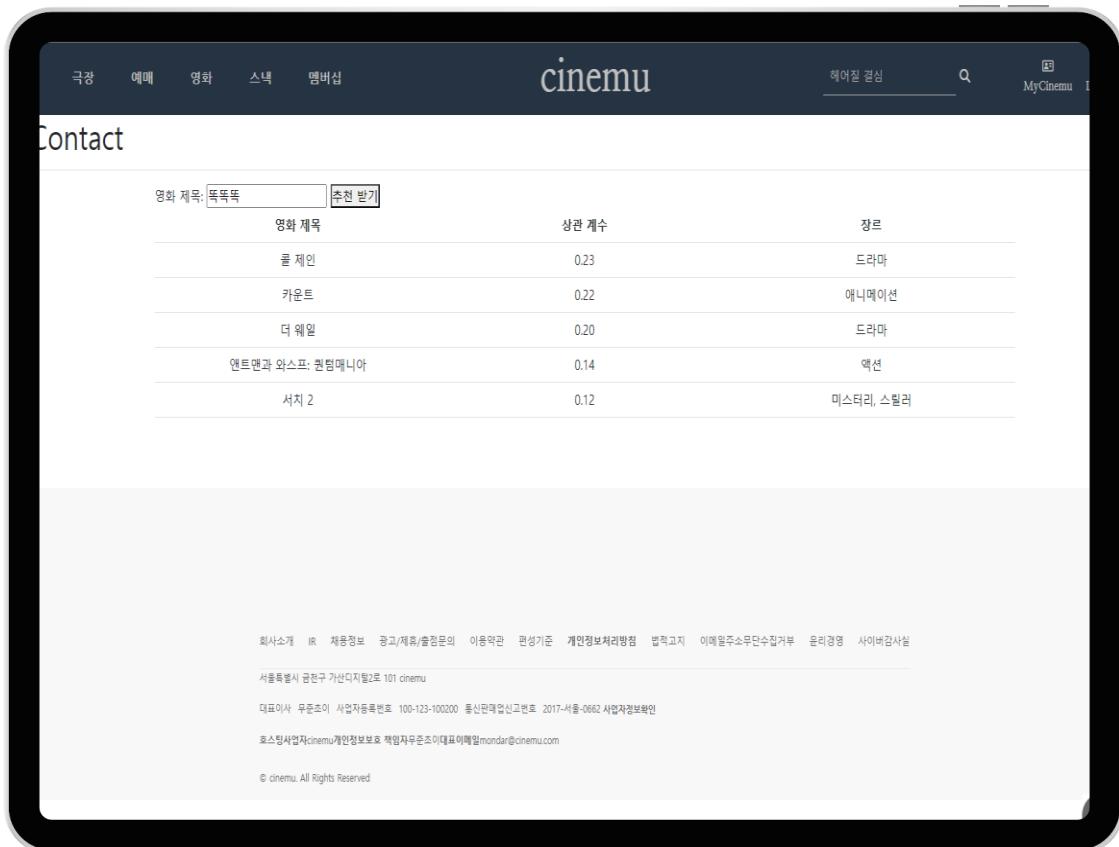
```
]     if request.method == 'GET':
        mname = request.GET.get('mname')
        if mname:
            recommend_result = recommend(mname, matrix, 5, similar_genre=True)
            result = pd.DataFrame(recommend_result, columns=['Title', 'Correlation', 'Genre'])
            response_data = {}
            response_data['result'] = result.to_dict(orient='records')

            # get 방식으로 전달되어 온 파라미터의 값
            json_callback = request.GET.get('callback')
            if json_callback:
                # callback(jsonData); 응답 객체를 임의로 만들어 줍니다.
                response = HttpResponse(
                    f'{json_callback}({json.dumps(response_data)}, safe=False);',
                    content_type="text/javascript; charset=utf-8"
                )
            else:
                response = JsonResponse(response_data, safe=False)
            return response
        else:
            return render(request, 'mvrd/mvrdlist.html')
    else:
        return render(request, 'mvrd/mvrdlist.html')
```



추천 – 영화 추천(cinemu)

결과 출력



```

<script>
$(document).ready(function() {
    $('#submit-btn').click(function() {
        let mname = $('input[name="mname"]').val();
        console.log('mname=>' + mname);
        $.ajax({
            type: 'GET',
            url: 'http://192.168.0.177:9000/mvrd/list',
            data: {
                'mname': $('input[name="mname"]').val(),
            },
            dataType: 'json',
            crossDomain: true,
            success: function(data) {
                console.log('JSONP');
                let recommendations = data.result;
                let resultList = $('#recommendation-list');
                resultList.empty();
                if (recommendations.length == 0) {
                    resultList.empty().append($('

').text('추천 결과가 없습니다.'));
                } else {
                    resultList.append($('

| ').text('영화 제목').appendTo(tr);                     \$(' <th>').text('상관 계수').appendTo(tr);                     \$('<th>').text('장르').appendTo(tr);                     let tbody = \$('<tbody>').appendTo(\$('#recommendation-list &gt; table'));                     \$.each(recommendations, function(i, recommendation) {                         let title = recommendation.Title;                         let correlation = recommendation.Correlation;                         let genre = recommendation.Genre.join(', ');                         let tr = \$('<tr>').appendTo(tbody);                         \$('<td>').text(title).appendTo(tr);                         \$('<td>').text(correlation).appendTo(tr);                         \$('<td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td></td></td></tr></tbody></th></th> | ').text('상관 계수').appendTo(tr);                     \$(' <th>').text('장르').appendTo(tr);                     let tbody = \$('<tbody>').appendTo(\$('#recommendation-list &gt; table'));                     \$.each(recommendations, function(i, recommendation) {                         let title = recommendation.Title;                         let correlation = recommendation.Correlation;                         let genre = recommendation.Genre.join(', ');                         let tr = \$('<tr>').appendTo(tbody);                         \$('<td>').text(title).appendTo(tr);                         \$('<td>').text(correlation).appendTo(tr);                         \$('<td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td></td></td></tr></tbody></th> | ').text('장르').appendTo(tr);                     let tbody = \$(' <tbody>').appendTo(\$('#recommendation-list &gt; table'));                     \$.each(recommendations, function(i, recommendation) {                         let title = recommendation.Title;                         let correlation = recommendation.Correlation;                         let genre = recommendation.Genre.join(', ');                         let tr = \$('<tr>').appendTo(tbody);                         \$('<td>').text(title).appendTo(tr);                         \$('<td>').text(correlation).appendTo(tr);                         \$('<td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td></td></td></tr></tbody> | ').text(title).appendTo(tr);                         \$(' <td>').text(correlation).appendTo(tr);                         \$('<td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td></td> | ').text(correlation).appendTo(tr);                         \$(' <td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td> | ').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$(' <p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ').text(title).appendTo(tr);                         \$(' <td>').text(correlation).appendTo(tr);                         \$('<td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td></td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | ').text(correlation).appendTo(tr);                         \$(' <td>').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$('<p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p></td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | ').text(genre).appendTo(tr);                     });                 }             },             error: function() {                 let resultList = \$('#recommendation-list');                 resultList.empty().append(\$(' <p>').text('추천 결과가 없습니다.'));             }         });     }); }); });</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                              |


```

PyCharm에서 보낸 데이터를 jsonp타입으로 받아서 출력

감사합니다!