
김예원

Developer

Git Hub : <https://github.com/wonnabe16>

Engineering Blog : <https://hailey-v.tistory.com>

E-mail : ywkim0822@naver.com

목차

Contents

1.

프로젝트 소개

Introduce Project

2.

구현 부분

Implement Part

3.

프로젝트 이미지

Project Image

4.

구현 방법

Implement Method

취미 상점

Hobby Market

참여 인원 : 6명

기여도 : 30%

진행 기간 : 2021.05.25-2021.07.07

프로젝트 설명

코로나 19로 인해 집에 머무는 시간이 길어진 현대인들이 새로운 취미를 배우는 등 자기 개발에 집중하는 사회 현상이 대두되었습니다. 이에 따라 재능을 가진 사람이 누구나 강사가 되어 소수 인원으로 배울 수 있는 원데이 클래스를 중개하는 웹 사이트를 제작하였습니다.

장점

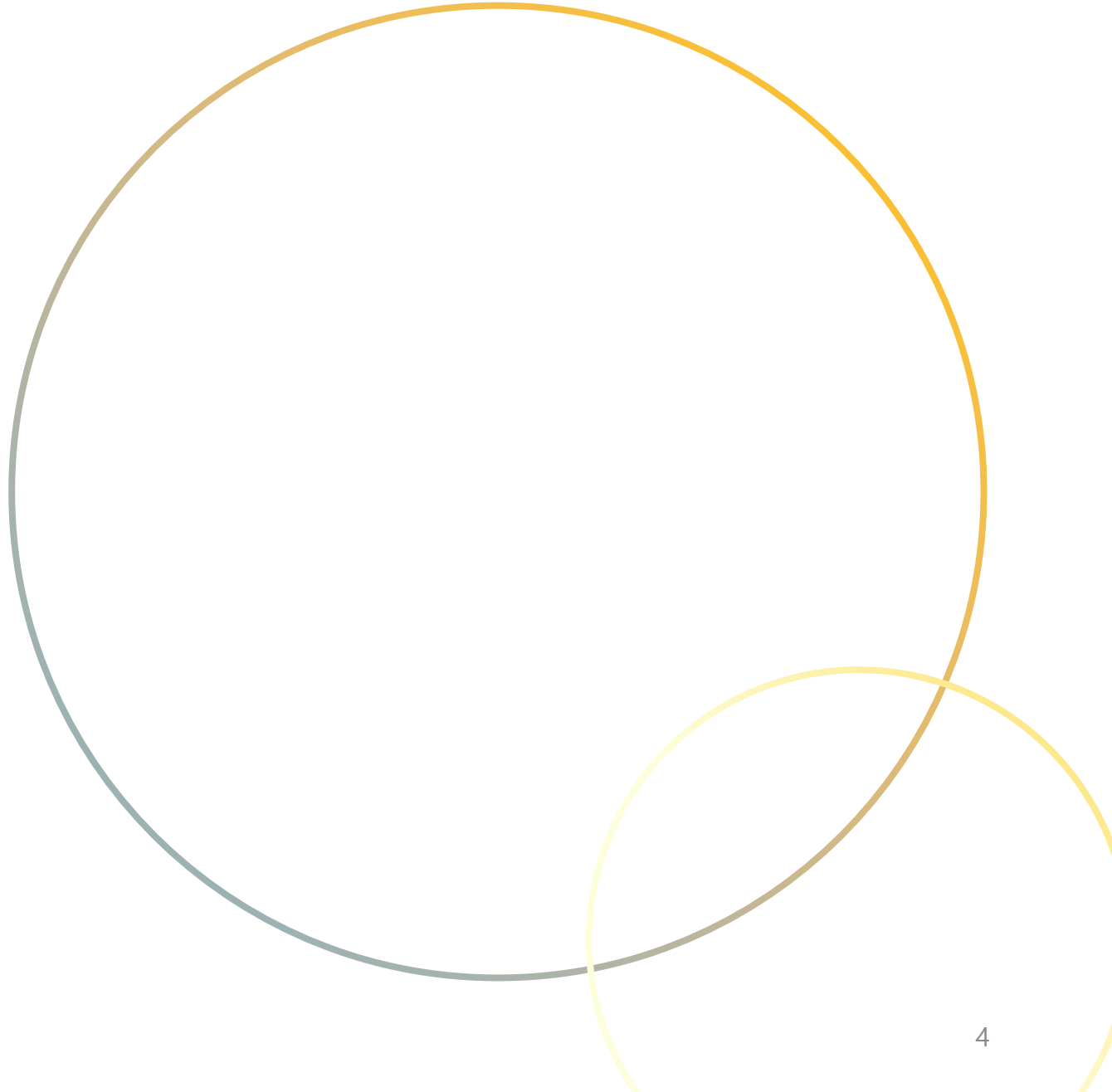
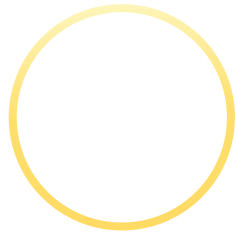
클래스를 등록하기 위한 간단한 검증 단계

취미를 수강할 수 있는 간단한 결제 단계

바람직한 커뮤니티를 위한 부적절한 콘텐츠를 관리

구현 부분

Implement part



구현 부분

Implement Part

취미 등록, 취미 목록

후기 등록, 수정, 조회

Front-end / Back-end

- Javascript - 클래스 조작
- Daum 주소 API
- 이미지 업로드 -
MultipartHttpServletRequest 클래스

- Modal을 사용한 게시글 등록
- 후기를 작성할 수 있는 필터 적용
(필터 : 해당 클래스를 결제한 사람,
여러 번 결제 이력이 있어도 후기는 하나
만 작성 가능하도록 구현)

프로젝트 이미지

Project Image

프로젝트 이미지

Project Image

취미 등록 page

클래스 기본정보

클래스 소개

클래스 일정 및 가격

클래스 추가정보

클래스 상세설명

클래스 등록 및 약관동의

취미 구매 옵션(1회권)

30000

취미 할인여부

할인 적용 ☒ (추후에 변경이 가능합니다.)

취미 할인율

1%

이전으로

다음으로

- 최소: 가로 600px X 세로 600px
- 권장: 가로 1000px X 세로 1000px (고해상도 지원)
- 1:1 비율의 (정사각형) 이미지를 권장합니다.
- 초상권, 저작권에 문제가 없어야 합니다.
- 상업적으로 사용 가능해야 합니다.
- 최소 1장 ~ 최대 5장까지 가능합니다.
- 용량 10MB 이하
- 포맷 jpg, png

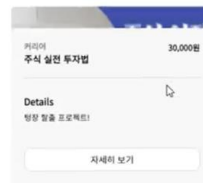


어떤 취미를 찾고 있나요?

로그인
아이디찾기

회원가입
비밀번호찾기

전체 카테고리 이벤트 오픈 예정 베스트 지원센터



취미 목록 page

프로젝트 이미지

Project Image

이벤트 오픈 예정 베스트 지원센터

후기5개 글쓰기

심지영
2021-07-25 16:05 작성됨
★★★★★

그냥 그랬어요ㅋㅋㅋㅋ 그래도 한 번은 재미로 해봐도 될 것 같아요~ㅋㅋㅋㅋ



도움데요 12

김연아
2021-07-25 16:04 작성됨
★★★★★

반신반의 했는데 금에달인거 맞았어요ㅋㅋㅋㅋ 감사합니다~



후기 작성
page

이벤트 오픈 예정 베스트 지원센터

후기 작성

내 평점 ★★★★★

너무 재밌어요~

파일 선택 선택된 파일 없음

제출하기

도움데요 14

후기 목록
page

프로젝트 이미지

Project Image

취미 별 후기 모아보기

 취미상점 상인

심지영님, 환영
합니다
로그아웃

취미관리

손님관리

정산

지원센터

마이페이지

후기

전체보기 검색

평균 평점 ★★★★★ 4.0



김사랑
2021-07-25 17:53 작성됨
★★★★★

너무 재밌어요~ 추천합니다!



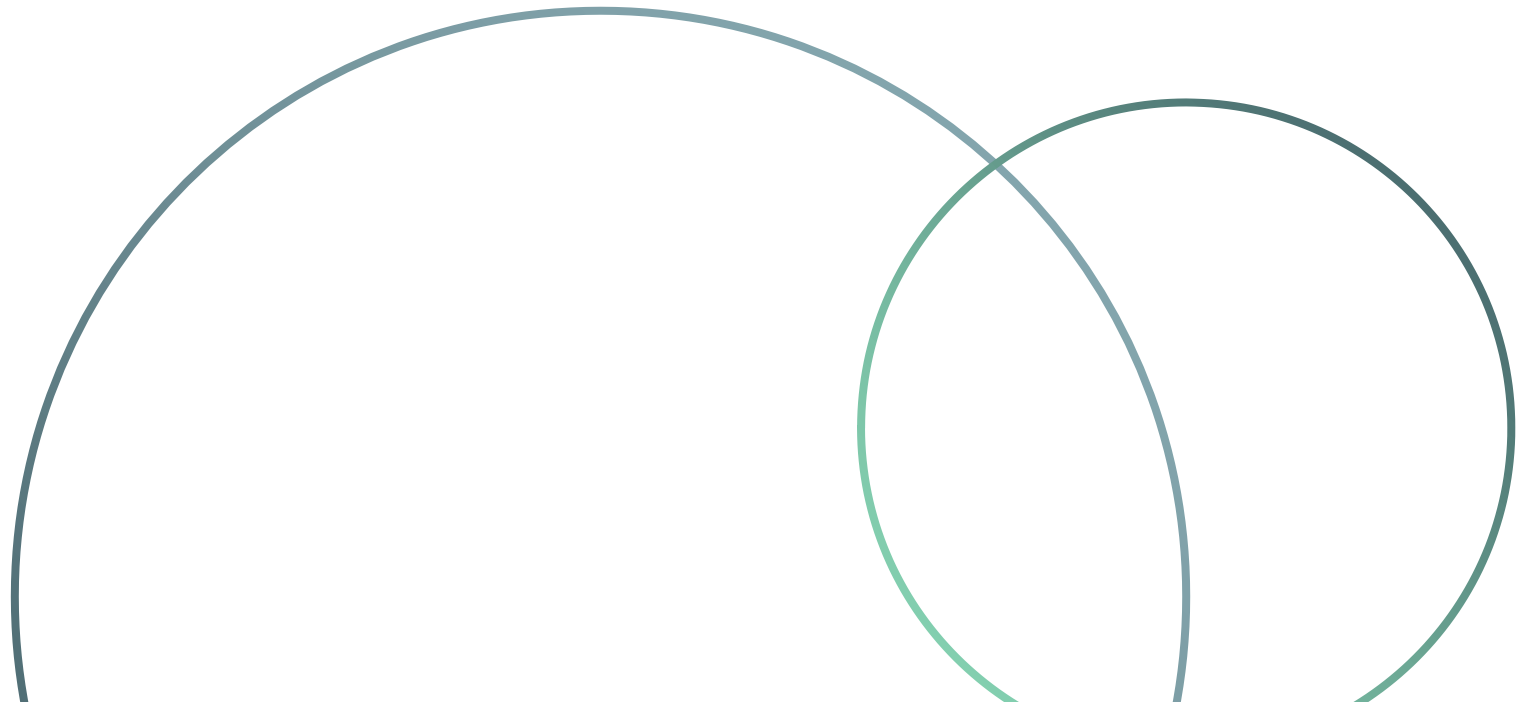
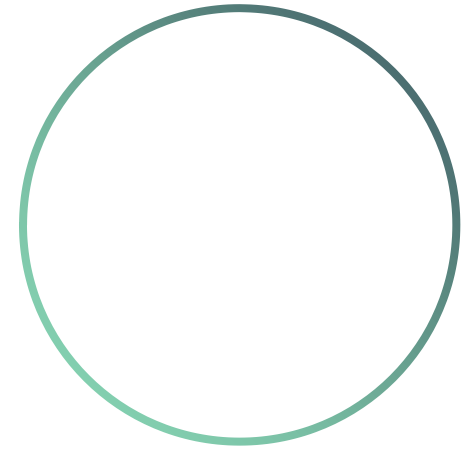
심지영
2021-07-25 16:05 작성됨
★★★★★

그냥 그랬어요ㅋㅋㅋㅋ 그래도 한 번은 재미로 해봐도 될 것 같아요~ㅋㅋㅋㅋ

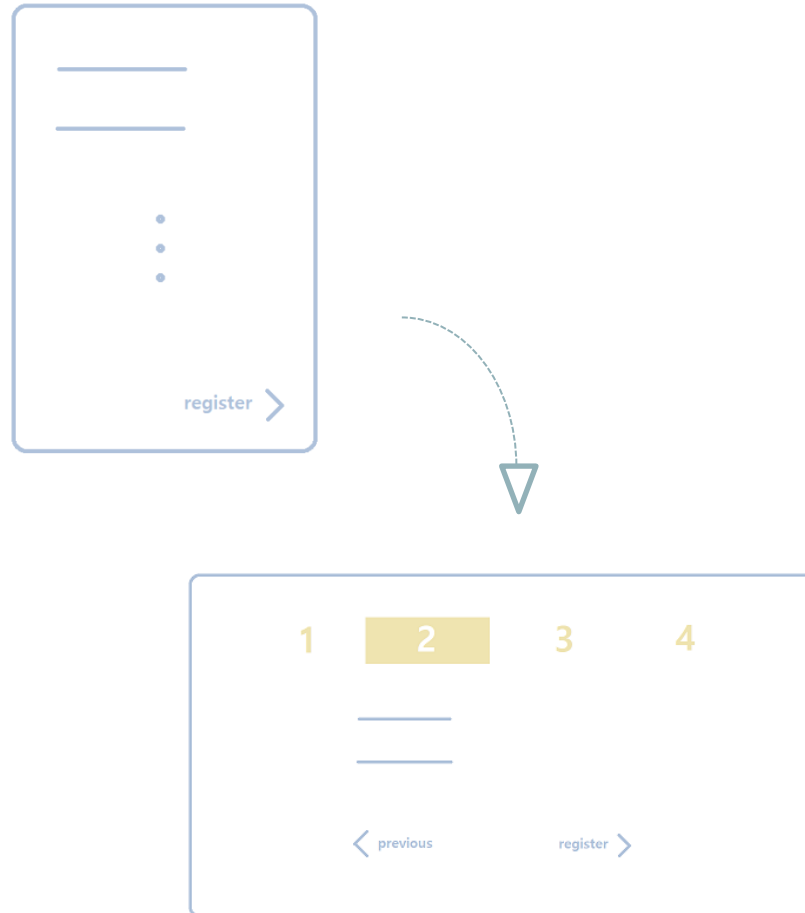


구현 방법

Implement Method



Javascript - 클래스 조작



문제

- 취미를 등록하기 위해 입력 받아야 할 정보가 굉장히 많음.
(이미지, 설명, 주소, 금액 등)

해결 방안

HTML

: 다수의 정보를 작은 단위로 입력가능하도록
form 태그 내부를 div로 Grouping

Javascript

: div에 active클래스를 추가, 제거 해줌으로써
단계간의 이동을 조작

Javascript - 클래스 조작

— 구현 화면

클래스 기본정보

클래스 소개

클래스 일정 및 가격

클래스 부가정보

클래스 상세설명

클래스 등록 및 약관동의

타이틀

취미 키워드를 적어주세요.(13자 이내)

간략소개

취미를 간략하게 소개해주세요.

카테고리

카테고리

다음으로

다음과 같은 경우 오픈이 어렵습니다.

- 소개팅 / 남녀 만남 주선 프리

- 드라이브, 안전이 보장되지 않은 장소에서 진행되는 프리

- 개인 판매 또는 타 플랫폼과 금액 차이가 나는 프리

- 상품설명이 300자 이내인 프리(띄어쓰기 포함)

- 프리이후, 추가 영입이 의심되는 상품 (보점,재무설계 등)

- 상품에 대한 아래사항 3가지 설명이 부족한 상품

① 프리 진행 방식

② 이 프리를 추천하고 싶은 사람들

③ 이 프리를 통해 대원이 얻어갈 수 있는 것

— 구현 코드

```
<div class="ul">
  <ul class="gnb">
    /* 다섯 단계 중 active 클래스가 붙은 요소가 화면에 표시됨 */
    <li class="step step-1 active"><span>클래스 기본정보</span></li>
    <li class="step step-2"><span>클래스 소개</span></li>
    <li class="step step-3"><span>클래스 일정 및 가격</span></li>
    <li class="step step-4"><span>클래스 부가정보</span></li>
    <li class="step step-5"><span>클래스 상세설명</span></li>
    <li class="step step-6"><span>클래스 등록 및 약관동의</span></li>
  </ul>
</div>

<div class="enroll enroll-1 active"></div>
<div class="enroll enroll-2"></div>
<div class="enroll enroll-3"></div>
<div class="enroll enroll-4"></div>
<div class="enroll enroll-5"></div>

<script>

  /* Javascript로 active 클래스 조작 */
  var seq = 1;
  $('<next>').click(function(){

    var is_empty = false;

    /* 다음 단계로 넘어가기 전 빈 값 체크 */
    $('<enroll->+seq).find('<req>').each(function(){
      if(!$(this).val()) {
        is_empty = true;
      }
    });
    if(is_empty) {
      alert('값을 전부 입력해주세요.');
```

이미지 업로드 - MultipartHttpServletRequest 클래스

— 구현 화면

클래스 기본정보

클래스 소개


클래스 일정 및 가격

클래스 부가정보

클래스 상세설명

클래스 등록 및 약관동의

대표 이미지





반려동물-펍네일3.png

업로드

추가 이미지

파일 선택

파일 2개



이전으로

다음으로

- 최소: 가로 600px X 세로 600px
- 권장: 가로 1000px X 세로 1000px (고해상도 지원)
- 1:1 비율의 (정사각형) 이미지를 권장합니다.
- 초상권, 저작권에 문제가 없어야 합니다.
- 상업적으로 사용 가능해야 합니다.
- 최소 1장 ~ 최대 5장까지 가능합니다.
- 용량 10MB 이하
- 포맷 jpg, png

— 구현 코드

```
/* 취미 등록 */
@PostMapping("/hobbyEnroll")
public ModelAndView enroll(ModelAndView model, @RequestParam("postcode") String postcode,
    @RequestParam("exactAddress") String exactAddress, MultipartHttpServletRequest mtfRequest,
    @SessionAttribute(name = "loginMerchant", required = false) Merchant loginMerchant,
    HttpServletRequest request, @ModelAttribute Hobby hobby) {

    List<String> setFile = null;

    /* 다중 파일 불러오기 */
    List<MultipartFile> fileList = mtfRequest.getFiles("file");

    /* 썸네일 파일 불러오기 */
    MultipartFile thumFile = mtfRequest.getFile("input-file");

    /* 상세페이지 이미지 파일 불러오기 */
    MultipartFile detailFile = mtfRequest.getFile("input-file-detail");

    /* 경로, 변수 설정 */
    String src = mtfRequest.getParameter("src");
    System.out.println("src value : " + src);
    String rootPath = mtfRequest.getSession().getServletContext().getRealPath("resources");
    String savePath = rootPath + "/upload/hobby/";

    /* 업로드된 파일이 있는 경우 이미지 저장 */
    if(!fileList.get(0).isEmpty()) {
        service.saveFiles(fileList, savePath, hobby);
    }
    if(!thumFile.isEmpty()) {
        setFile = service.saveFile(thumFile, savePath, hobby);
        hobby.setHbThumOri(setFile.get(0));
        hobby.setHbThumRename(setFile.get(1));
    }
    if(!detailFile.isEmpty()) {
        setFile = service.saveFile(detailFile, savePath, hobby);
        hobby.setHbDetailOri(setFile.get(0));
        hobby.setHbDetailRename(setFile.get(1));
    }

    . . .

    return model;
}
```

Daum 주소 API

— 구현 화면

태해란로14길

검색결과가 많습니다. 검색어에 아래와 같은 조합을 이용하시면 더욱 정확한 결과가 검색됩니다. '도로명+건물번호', '지역명+지번', '지역명+건물명(아파트명)', '사서함명+번호'

06235 영문보기 지도
도로명 서울 강남구 테헤란로14길 5 (삼성역상빌딩)
지번 서울 강남구 역삼동 735-10

06234 영문보기 지도
도로명 서울 강남구 테헤란로14길 6 (남도빌딩)
지번 서울 강남구 역삼동 823-24
노외지번주소 1건 (더보기)

06234 영문보기 지도
도로명 서울 강남구 테헤란로14길 8 (강남역두산위브세션룸)
지번 서울 강남구 역삼동 823-26

주소 찾기

상세주소

이전으로 등록하기

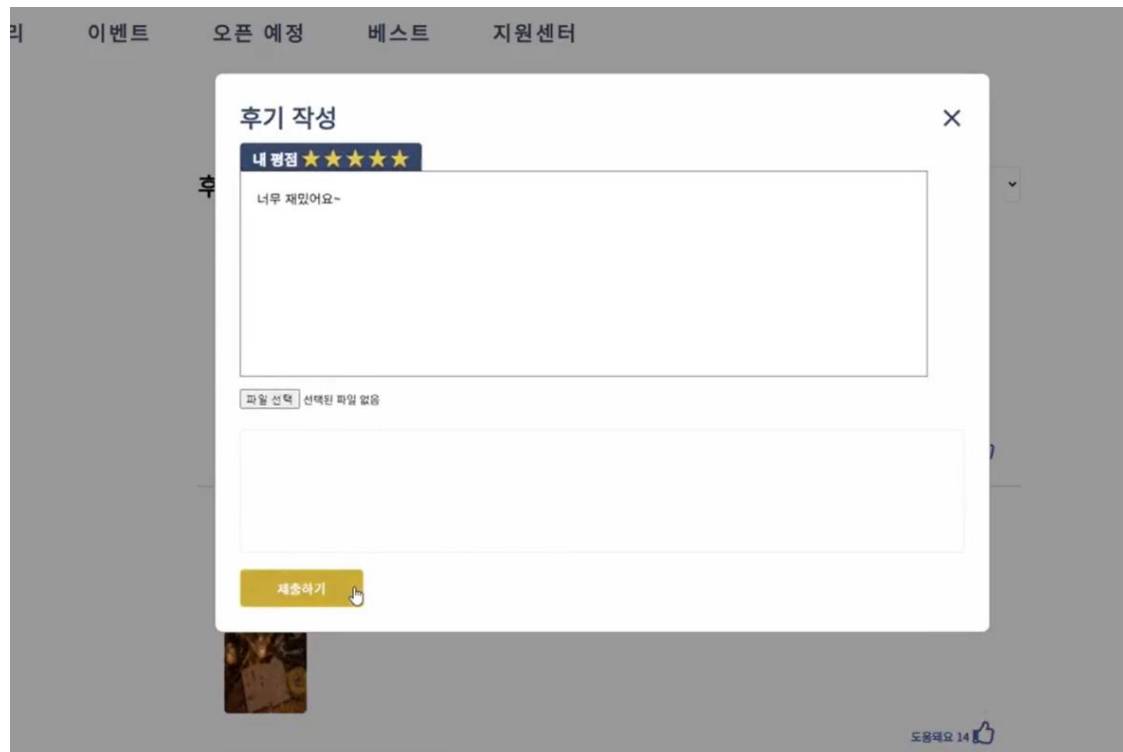
— 구현 코드

```
/* Daum 주소 API */
function sample4_execDaumPostcode() {
    new daum.Postcode({
        onComplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.
            // 도로명 주소의 노출 규칙에 따라 주소를 표시한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var roadAddr = data.roadAddress; // 도로명 주소 변수
            var extraRoadAddr = ''; // 참고 항목 변수
            // 법정동명이 있을 경우 추가한다. (법정리는 제외)
            // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
            if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                extraRoadAddr += data.bname;
            }
            // 건물명이 있고, 공동주택일 경우 추가한다.
            if(data.buildingName !== '' && data.apartment === 'Y'){
                extraRoadAddr += (extraRoadAddr !== ' ? ' ? ' ' + data.buildingName : data.buildingName);
            }
            // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
            if(extraRoadAddr !== ''){
                extraRoadAddr = ' (' + extraRoadAddr + ')';
            }
            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById('postcode').value = data.zonecode;
            document.getElementById('roadAddress').value = roadAddr;
            /* document.getElementById('sample4_jibunAddress').value = data.jibunAddress; */
            child.close();

            // 참고항목 문자열이 있을 경우 해당 필드에 넣는다.
            if(roadAddr !== ''){
                document.getElementById("exactAddress").value = extraRoadAddr;
            } else {
                document.getElementById("exactAddress").value = '';
            }
            var guideTextBox = document.getElementById("guide");
            // 사용자가 '선택' 인원을 클릭한 경우, 예상 주소라는 표시를 해준다.
            if(data.autoRoadAddress) {
                var expRoadAddr = data.autoRoadAddress + extraRoadAddr;
                guideTextBox.innerHTML = '(예상 도로명 주소 : ' + expRoadAddr + ')';
                guideTextBox.style.display = 'block';
            } else if(data.autoJibunAddress) {
                var expJibunAddr = data.autoJibunAddress;
                guideTextBox.innerHTML = '(예상 지번 주소 : ' + expJibunAddr + ')';
                guideTextBox.style.display = 'block';
            } else {
                guideTextBox.innerHTML = '';
                guideTextBox.style.display = 'none';
            }
        }
    }).open();
}
```

Modal을 사용한 게시물 등록

구현 화면



구현 코드

```
<script>

/* 후기 글 수정 관련 modal */
var modal = document.getElementById('myModalupd');
var btn = document.getElementById("update");
var span = document.getElementsByClassName("close")[0];

/* 후기 글쓰기 버튼을 클릭하면 모달창이 나타남 */
btn.onclick = function() {
    modal.style.display = "block";
}

/* X버튼을 누르면 모달창이 사라짐 */
span.onclick = function() {
    modal.style.display = "none";
}

/* 모달창 외부 클릭하면 모달창이 사라짐 */
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}

</script>
```

후기를 작성할 수 있는 필터 적용
(구현하면 위와 동일)

— 구현 코드

```
/* 취미 후기페이지 요청 */
@GetMapping("/view/reviewList")
public ModelAndView reviewList(ModelAndView model,
    @RequestParam(value = "page", required = false, defaultValue = "1") int page,
    @SessionAttribute(name = "loginMember", required = false) Member loginMember,
    @RequestParam("hbNo") int hbNo) {

    . . .

    /* 수강 이력이 있고, 후기를 한번도 작성하지 않았다면 글쓰기 가능 */
    if (loginMember != null && service.reviewFindByNo(hbNo, loginMember.getMemNo()) == null) {

        int memNo = loginMember.getMemNo();
        int reserveCount = service.getReserveCount(hbNo, memNo);
        model.addObject("reserveCount", reserveCount);
    }

    /* 수강 이력이 있고, 이미 작성한 후기가 있다면 해당 후기 정보 가져오기 */
    if (loginMember != null && service.reviewFindByNo(hbNo, loginMember.getMemNo()) != null) {
        int memNo = loginMember.getMemNo();
        Review reviewByNo = service.reviewFindByNo(hbNo, memNo);
        model.addObject("reviewByNo", reviewByNo);
    }

    . . .

    return model;
}
```