# SANGHYANG

# 음 팀원 소개







이승요



윤주영



김채은



김초원



김예빈



김지민

#### 목차 Contents

01 프로젝트 개요

02 계획 및 방법

03 결과 및 시연

04 개선점 및 느낀 점

### 프로젝트 개요



#### 理想鄉

[명사] 인간이 생각할 수 있는 최선의 상태를 갖춘 완전한 사회

[유의어] 낙원, utopia

#### 理想香

"가장 완전하다고 여겨지는 향기"

















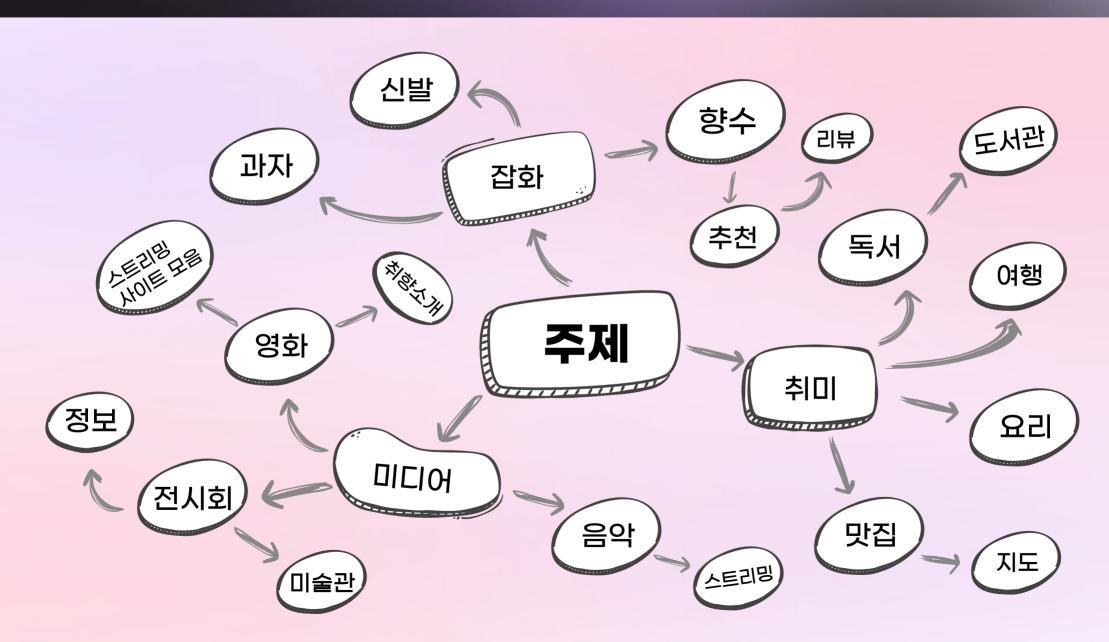














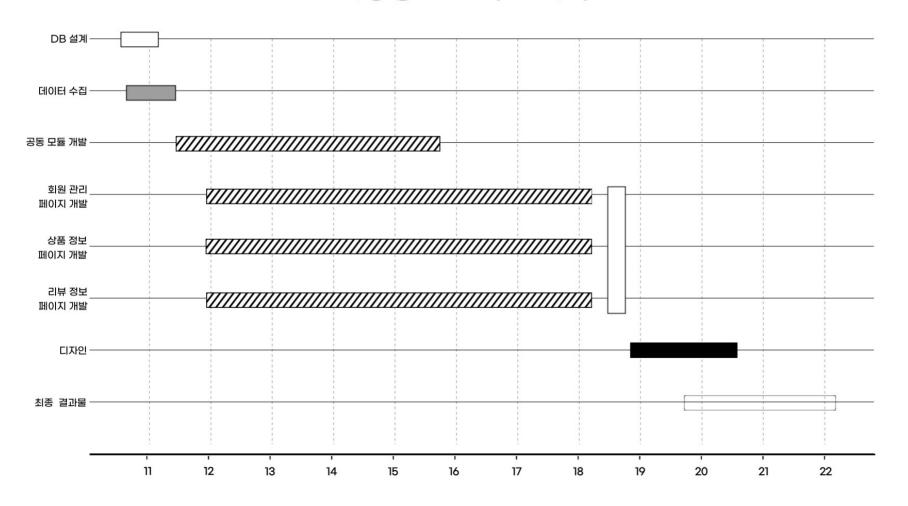
CRUD 기능을 구현하며 MySQL DB를 활용해 데이터를 저장하고 저장한 데이터를 출력하는 기능을 최대한 활용할 수 있는

다양한 브랜드, 종류를 포함하고 있는 향수를 소개하고 소통하는 커뮤니티 홈페이지를 제작하기로 결정



# 계획 및 방법

#### 이상향 프로젝트 계획

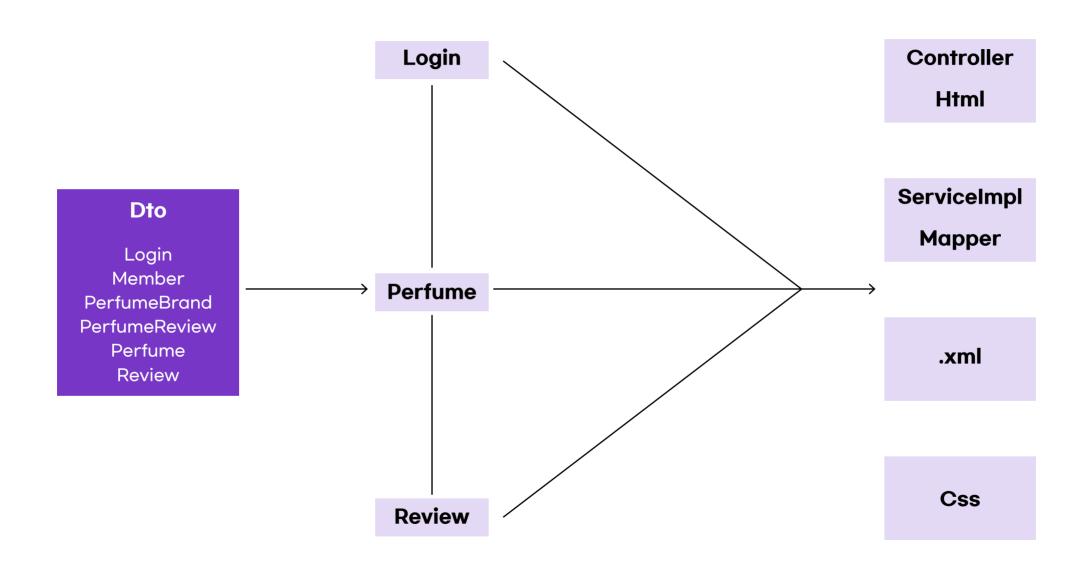


#### 기능별 역할 분담

**회원 관리 개발** 문기은, 김예빈

**제품 목록 개발** 윤주영, 이승요, 김지민

**리뷰 커뮤니티 개발** 김초원, 김채은



#### 개발환경







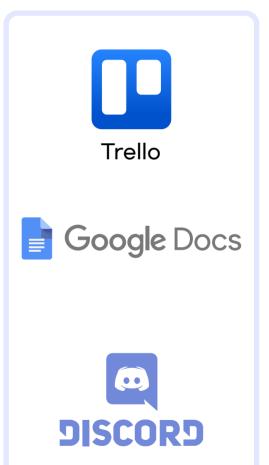
#### 사용언어



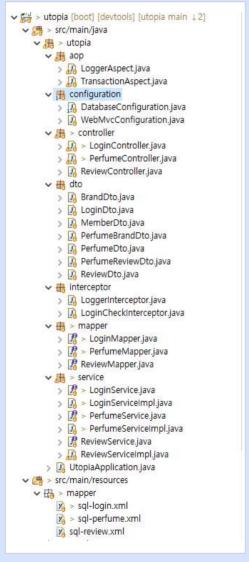
#### 형상관리&디자인

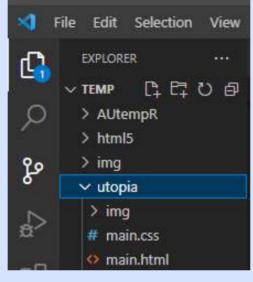


#### 협업



#### 개발환경과 사용언어

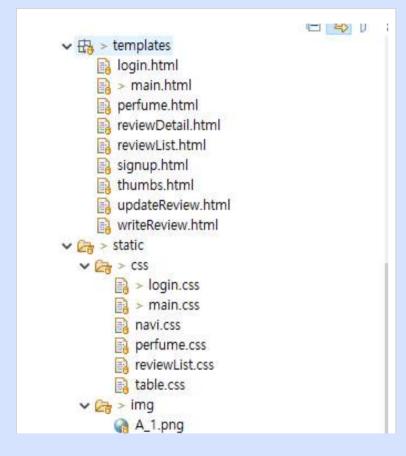




#### Github 형상관리



#### 디자인 html, css



#### 이미지 파일명

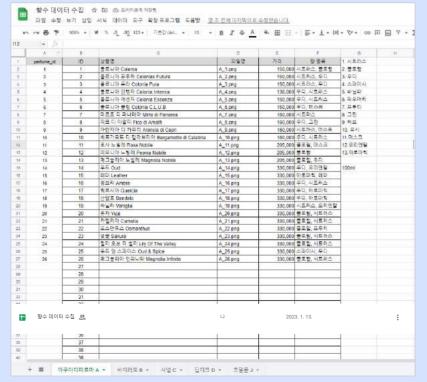
<b>■</b> A_1	■ A_25	■ B_23	<b>■</b> C_21	■ D_6	■ J_4
■ A_2	■ A_26	■ B_24	<b>■</b> C_22	<b>■</b> D_7	■ J_5
<b>■</b> A_3	■ B_1	■ B_25		■ D_8	■ J_6
■ A_4	■ B_2	■ B_26		■ D_9	<b>■</b> J_7
■ A_5	■ B_3			■ D_10	<b>■</b> J_8
■ A_6	<b>■</b> B_4		<b>■</b> C_26	■ D_11	■ J_9
<b>■</b> A_7	■ B_5		C_30	■ D_12	■ J_10
<b>■</b> A_8	■ B_6	C_4	C_31	■ D_13	■ J_11
■ A_9	■ B_7			■ D_14	■ J_12
■ A_10	■ B_8	<b>■</b> C_6	<b>■</b> C_33	■ D_15	■ J_13
■ A_11	■ B_9		C_34	■ D_16	■ J_14
■ A_12	■ B_10	<b>■</b> C_8	<b>■</b> C_35	■ D_17	■ J_15
■ A_13	■ B_11		<b>■</b> C_36	■ D_18	■ J_16
■ A_14	■ B_12		<b>■</b> C_37	■ D_19	■ J_17
■ A_15	■ B_13	■ C_11	<b>■</b> C_38	■ D_20	■ J_18
■ A_16	■ B_14		<b>■</b> C_39	■ D_21	■ J_19
■ A_17	■ B_15	C_13	C_40	■ D_22	■ J_20
■ A_18	■ B_16	C_14	C_41	■ D_23	■ J_21
■ A_19	■ B_17		C_42	■ D_24	
■ A_20	■ B_18		■ D_1	■ D_25	
■ A_21	■ B_19	■ C_17	■ D_2	■ D_26	
■ A_22	■ B_20	■ C_18	■ D_3	■ J_1	
A_23	■ B_21		<b>■</b> D_4	■ J_2	
■ A_24	■ B_22	C_20	■ D_5	■ J_3	

login_bg
🔊 login_logo
🔳 logo
logo_text
logoW
main_bg2
main3
mainunder_txt
product_bg
quickmenu1
quickmenu2
quickmenu3
quickmenu4
quickmenu5
quickmenu6
review_bg
🖹 signup_bg
🗃 signup_logo
curation1
curation2
curation3

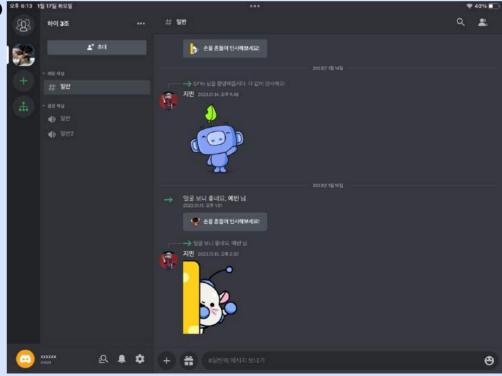
#### **Trello**

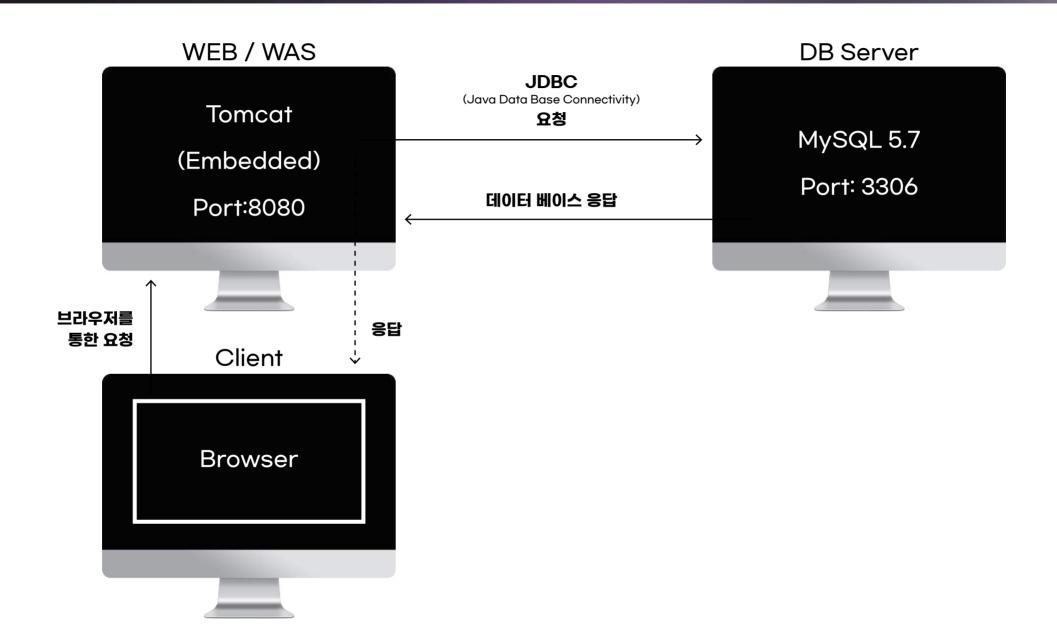


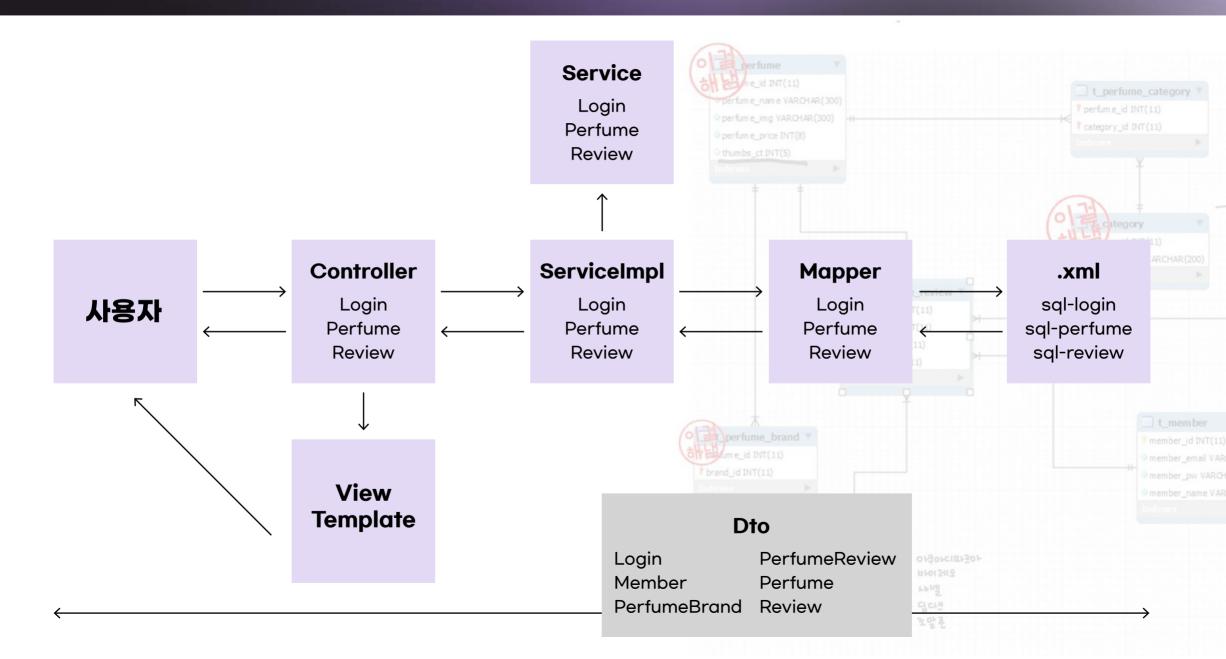
#### 구글 공유 문서



#### **DISCORD**









#### preHandler

- LoggerInterceptor
- LoginCheckInterceptor

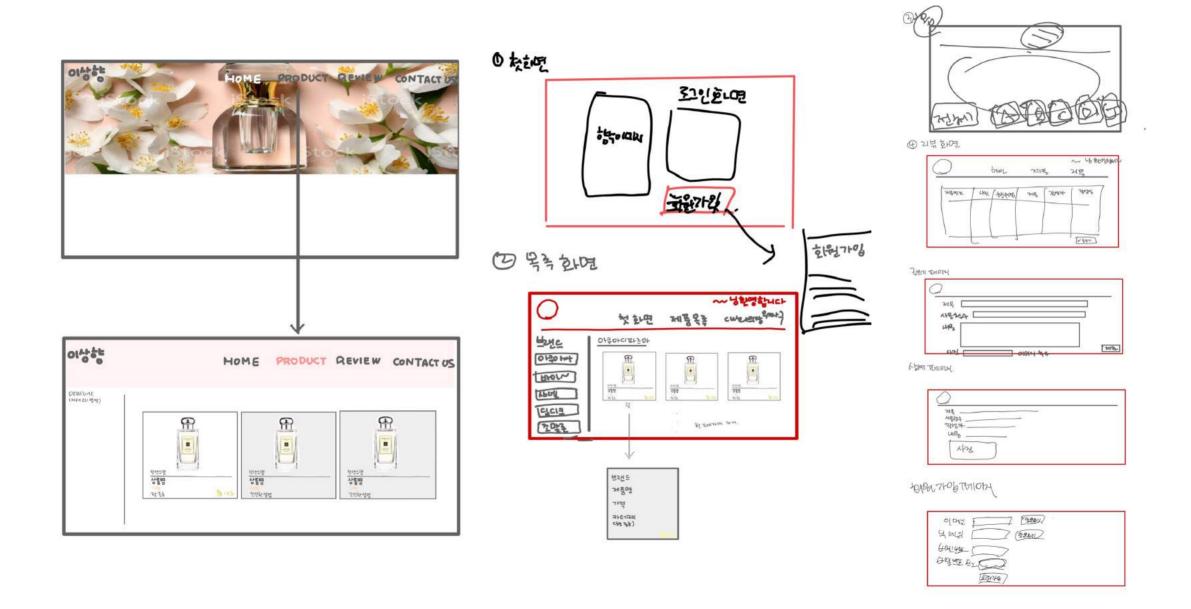
#### Dispatcher Servlet

- LoggerInterceptor

postHandler

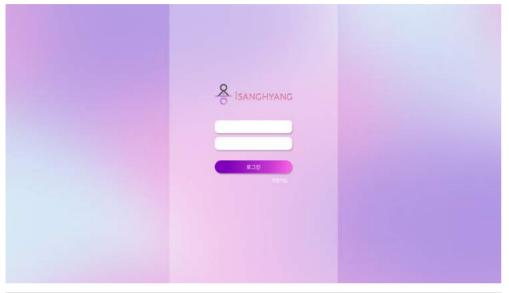
Interceptor

Controller

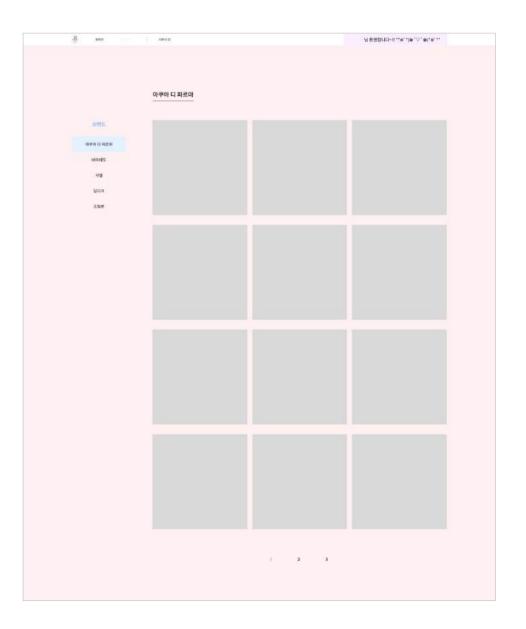


02 계획 및 방법 화면설계 - 피그마

#### 로그인, 메인, 제품 목록 설계







#### 리뷰 커뮤니티 게시판 (상세조회, 등록, 수정 )

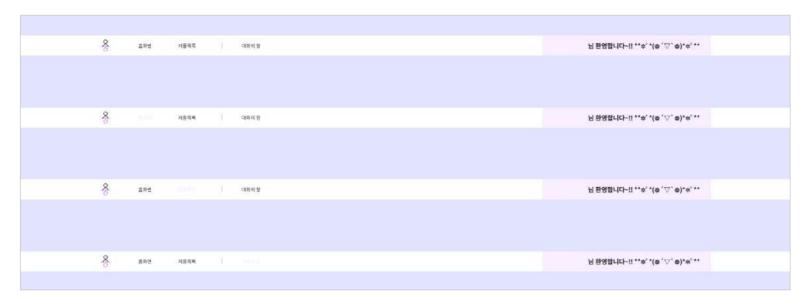






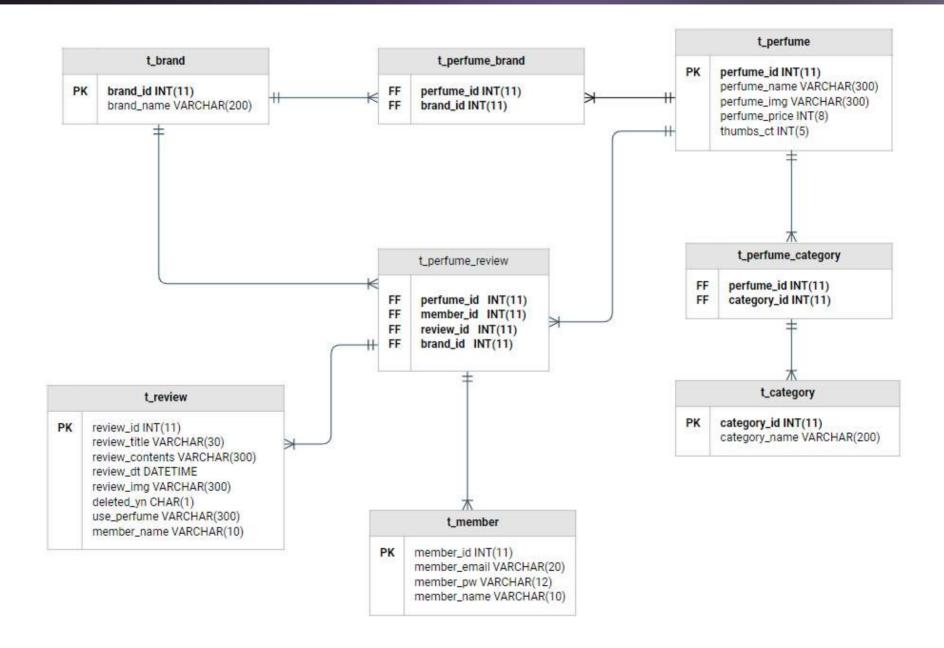


#### 네비게이션 바, 등록, 수정, 삭제 버튼, 사이드바 메뉴 정의





브랜드	브랜드	브랜드	브랜드	브랜드	브랜드
아쿠아 디 파르마					
바이레도	바이레도	바이레도	바이레도	바이레도	바이레도
샤넬	샤넬	샤넬	샤넬	샤넬	샤넬
딥디크	딥디크	딥디크	딥디크	딥디크	딥디크
조말론	조말론	조말론	조말론	조말론	조말론



#### 02 계획 및 방법

column_name	data_type	column_type	column_key	is_nullable	column_default	extra	comment
brand_id	int	int(11)	PRI	NO		auto_increment	
brand_name	varchar	varchar(200)		NO			
category_id	int	int(11)	PRI	NO		auto_increment	
category_name	varchar	varchar(200)		NO			
member_id	int	int(11)	PRI	NO		auto_increment	
member_email	varchar	varchar(20)		NO			
member_pw	varchar	varchar(12)		NO			
member_name	varchar	varchar(10)		NO			
perfume_id	int	int(11)	PRI	NO		auto_increment	향수 ID 등록 순으로 1씩 증가함
perfume_name	varchar	varchar(300)		NO			향수 이름
perfume_img	varchar	varchar(300)		NO			향수 이미지
perfume_price	int	int(8)		NO			향수 가격
thumbs_ct	int	int(5)		YES	0		따봉 갯수
perfume_id	int	int(11)	PRI	NO			
brand_id	int	int(11)	PRI	NO			
perfume_id	int	int(11)	PRI	NO			
category_id	int	int(11)	PRI	NO			
perfume_id	int	int(11)	PRI	NO			
member_id	int	int(11)	PRI	NO			
review_id	int	int(11)	PRI	NO			
brand_id	int	int(11)	PRI	NO			
review_id	int	int(11)	PRI	NO		auto_increment	
review_title	varchar	varchar(30)		NO			
review_contents	varchar	varchar(300)		NO			
review_dt	datetime	datetime		NO	CURRENT_TIMESTAMP		
review_img	varchar	varchar(300)		YES			
deleted_yn	char	char(1)		NO	N		
use_perfume	varchar	varchar(300)		YES			
perfume_brand	varchar	varchar(300)		YES			
member_name	varchar	varchar(10)		NO			

## 결과 및 시연

03 결과 및 시연

#### LoginCheckInterceptor.java

```
public class LoginCheckInterceptor implements HandlerInterceptor {
            @Override
            public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
            if (request.getRequestURI().indexOf("/utopia") >= 0 && request.getSession().getAttribute("member") == null) {
                        request.getSession().setAttribute("message", "로그인 후 사용하실 수 있습니다.");
                        response.sendRedirect("/login.do");
                        return false; }
            else {
                        return true; }
```

LoginCheckInterceptor.java

#### request.getRequestURI().indexOf("/utopia") >= 0

ablic class LoginCheckInterceptor implements HandlerInterceptor (

&& request.getSession().getAttribute("member") == null

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception

```
If (request.getRequestURIO.indexOit"/uropia") >= 0 전환 request.getSession().getAttribute("member") == null) (
request.getSession().setAttribute("message", "로그인 후 사용하실 수 있습니다."):
response, sendRedirect("/login,do");
return false: )
else (
return true; )
```

/utopia/login.do로 접속했을 때 뜨는 초기 오류 화면



로그인 체크 인터셉터에서 로그인 페이지로 리다이렉팅되는 조건 if (request.getRequestURI().indexOf("/utopia") >= 0 &&

request.getSession().getAttribute("member") == null)

```
@PostMapping("/utopia/login.do")
public String login(LoginDto loginDto, HttpSession session) throws Exception {
    MemberDto memberDto = loginService.login(loginDto);

if (memberDto == null) {
        session.setAttribute("message", "일치하는 정보가 존재하지 않습니다.");
        return "redirect/login.do"; }

else {
        session.setAttribute("member", memberDto);
        return "redirect/main.do"; }
}
```



```
@PostMapping("/login.do")
public String login(LoginDto loginDto, HttpSession session) throws Exception {
    MemberDto memberDto = loginService.login(loginDto);

if (memberDto == null) {
        session.setAttribute("message", "일치하는 정보가 존재하지 않습니다.");
        return "redirect/login.do"; }

else {
        session.setAttribute("member", memberDto);
        return "redirect/main.do"; }
}
```

```
/utopia/login.do
                       Mys Exception (
    login.do
```

03 결과 및 시연

#### LoginController.java

```
<script type="text/javascript">
      function passConfirm() {
            var password = document.getElementById('memberPw');
            var passwordConfirm = document.getElementByld('memberPwCheck');
            var confirmMsg = document.getElementById('confirmMsg');
            var correctColor = "green";
            var wrongColor = "#ff0000";
            if (password.value === passwordConfirm.value) {
                   confirmMsg.style.color = correctColor
                   confirmMsg.innerHTML ="비밀번호 일치 }
            else {
                   confirmMsg.style.color = wrongColor;
                   passwordConfirm.focus();
                   confirmMsg.innerHTML ="비밀번호 불일치"; }
</script>
```

#### LoginController.javo

```
password.value === passwordConfirm.value
```

#### PerfumeController.java

```
@GetMapping("/utopia/perfume")
public ModelAndView perfume(
@RequestParam(value = "brandId", required = false, defaultValue = "0") int brandId,
@RequestParam(value = "currentPage", required = false, defaultValue="1") int currentPage) throws Exception {
ModelAndView mv = new ModelAndView("perfume.html");
if (brandId == 0) {
       List < PerfumeDto > list = perfumeService.selectPerfumeList((currentPage - 1) *30 );
       mv.addObject("perfumeList", list);
       mv.addObject("pageCount", Math.cei/(perfumeService.selectPerfumeListCount() / 30.0));
       mv.addObject("currentPage", currentPage); }
else {
       List < PerfumeDto > list = perfumeService.selectBrandPerfumeList(brandId);
       mv.addObject("perfumeList", list); }
List < BrandDto > brandDto = perfumeService.selectBrandList();
mv.addObject("brandList", brandDto);
return mv; }
```

PerfumeController java

@RequestParam(value = "brandId", required = false, defaultValue = "0")

if (brandId == 0) {

List < PerfumeDto > list = perfumeService.selectPerfumeList((currentPage - 1) \*30 );

nw.addObject(ipage\_ount), Math.ce/(pertumes/eviceselectPertumeListCount)
mv.addObject(irumentPane), purrentPane)

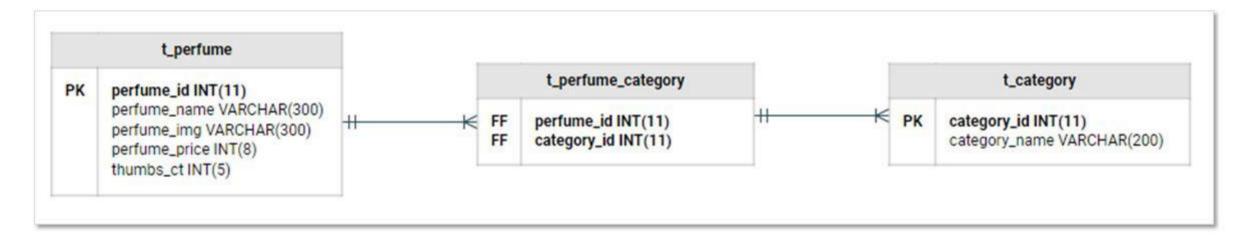
else {

List < PerfumeDto > list = perfumeService.selectBrandPerfumeList(brandId);

. List<BrandDto> brandDto = perfumeServices.electBrandListO

mv.addObject("brandUst", brandDto);

#### perfume 테이블과 category 테이블을 맵핑하는 테이블 구조



```
@GetMapping("/utopia/perfume")
public ModelAndView perfume(
@RequestParam(value = "brandId", required = false, defaultValue = "0") int brandId) throws Exception {
ModelAndView mv = new ModelAndView("perfume.html");
if (brandId == 0) {
      List < PerfumeDto > list = perfumeService.selectPerfumeList();
      mv.addObject("perfumeList", list);
else {
      List < PerfumeDto > list = perfumeService.selectBrandPerfumeList(brandId);
      mv.addObject("perfumeList", list); }
List < BrandDto > brandDto = perfumeService.selectBrandList();
mv.addObject("brandList", brandDto);
return mv; }
```

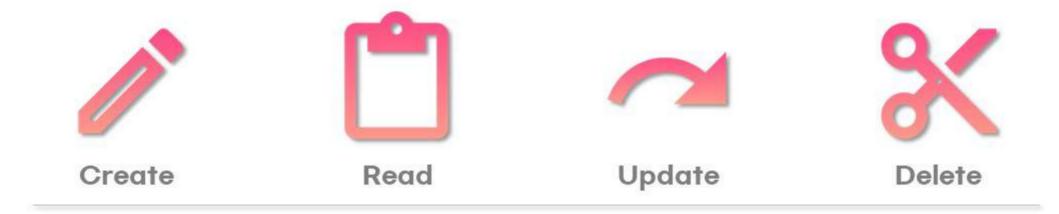
```
defaultValue = "0") int brandId)
```

@RequestParam(value = "brandId", required = false,

```
@GetMapping("/utopia/perfume")
public ModelAndView perfume(
@RequestParam(value = "brandId", required = false, defaultValue = "0") int brandId,
@RequestParam(value = "currentPage", required = false, defaultValue="1") int currentPage) throws Exception {
ModelAndView mv = new ModelAndView("perfume.html");
if (brandId == 0) {
      List < PerfumeDto > list = perfumeService.selectPerfumeList((currentPage - 1) *30 );
      mv.addObject("perfumeList", list);
      mv.addObject("pageCount", Math.cei/(perfumeService.selectPerfumeListCount() / 30.0));
      mv.addObject("currentPage", currentPage);
} else {
      List < PerfumeDto > list = perfumeService.selectBrandPerfumeList(brandId);
      mv.addObject("perfumeList", list); }
      List < BrandDto > brandDto = perfumeService.selectBrandList();
      mv.addObject("brandList", brandDto);
return mv; }
```

```
@RequestParam(value = "currentPage", required = false,
                                defaultValue="1") int currentPage)
```

03 결과 및 시연



**CRUD** 

```
<select id="selectReviewByReviewId" parameterType="int" resultType="utopia.dto.ReviewDto">
        SELECT review_img
        FROM t_review
        WHERE review_id = #{reviewId}
</select>
```



```
<select id="selectReviewByReviewId" parameterType="int" resultType="utopia.dto.ReviewDto">
        SELECT review_id, review_title, review_contents, review_img, use_perfume, perfume_brand
        FROM t_review
        WHERE review_id = #{reviewId}
    </select>
```



WHERE review Id = #{reviewId}

</select>



</select>

#### UpdateReview.html

```
제목
         <input class= "input" type= "text" th:value= "${review.reviewTitle}" id= "reviewTitle" name= "reviewTitle" />
    사용 향수
         <input class= "input" type= "text" th:value= "${review.usePerfume}" id= "usePerfume" name= "usePerfume" />
    내용
         <textarea class= "textarea" id= "reviewContents" th:text= "${review.reviewContents}" name= "reviewContents"></textarea>
```

```
th:value= "${review.reviewTitle}"
                                      GUTTLE" />c/td>
th:value="${review.usePerfume}"
th:text="${review.reviewContents}"
```

## 개선점 및 느낀 점

### 프로젝트를 마치며 아쉬운 점이 있나요?? 🤔 🤥 🤥







카테고리별 출력하는 기능을 구현하지 못한 것이 아쉬워요

브랜드, 멤버, 향수, 리뷰를 모두 맵핑하는 PerfumeReview 테이블을 활용하지 못한 점이 아쉽습니다

회원 가입 시 이메일 중복값을 체크하는 기능을 구현하지 못했어요



깃허브를 사용한 형상 관리에 어려움이 있었습니다

#### 프로젝트를 마치며 팀원들의 느낀 점을 말해주세요~! 😎 😎









프로젝트를 진행 함으로써까지 배웠던 내용들에 대해 자세히 복습하는 시간을 갖게 되었고 조원들이 다들 맡은 역할을 제대로 해주어서 프로젝트 진행이 원활하게 되어 좋았습니다



페이지 하나를 구성하는 데에 있어 필요한 점이 얼마나 많은지 직접 구현하면서 깨달을 수 있었습니다. 또한 협업을 통해 구현할 수 없을 것 같은 부분도 무리 없이 진행할 수 있었고, 여러 가지 아이디어로 더욱 좋은 방법을 찾을 수 있어서 좋았습니다.



CRUD의 흐름과, 수업 시간에 배웠던 내용들을 다시 한번 복습할 수 있는 시간을 보내서 좋았습니다.



코드를 따라 치기 바빴던 한 달 차에 시작한 프로젝트라 걱정이 많았지만 생각보다 좋은 팀원들 덕분에 수월하다고 느껴질 만큼 잘 진행되었습니다 프로젝트를 진행하면서 CRUD에 대해 이해하고 조금은 오류코드를 더 잘 해석할 수 있게 됐다고 생각합니다 이외 에도 이해하지 못하고 썼던 코드들이 조금은 이해가 되고 팀원들과 함께 고민하고 해결책을 찾으면서 배우는 점이 많았습니다 아쉬운 점은 있지만 화면 설계보다는 완성도 있게 만든 것 같아 좋았습니다



페이지의 기능 구현을 위해 백엔드와 프론트엔드 어느 하나 쉬운 것이 없었지만, 부족한 부분은 팀원들과 함께 해결했기에 결과물을 만들어 낼 수 있었던 것 같습니다. 물론 처음 회의에서 나왔던 아이디어 중에 구현하지 못한 기능들이 있어 아쉬운 부분도 있지만, 팀원 간의 협업이 잘 이루어져서 만족스러운 결과물을 만들어 낼 수 있었고, 프론트와 백엔드의 전체적인 흐름을 조금이나마 파악할 수 있었던 좋은 기회였습니다.



팀 프로젝트 과정에서 웹 개발 지식과 기술, 의사소통 능력 등을 배우고, 개선점을 보완하고자 하는 열정과 결과물에 대한 성취감을 고취시킬 수 있었습니다. 아직 부족한 점이 많지만 이러한 기회를 통해서 꾸준히 성장할 수 있는 개발자가 되고 싶습니다.



프로젝트가 시작했을 때, 평소 실력에 부족함을 늘 느끼던 저는 막막했습니다. 하지만 막상 첫 회의를 시작하고, 팀원분들과 함께 프로젝트의 방향성을 정하고, 한 줄씩 코드를 직접 짜보기 시작하니 즐겁게 몰두할 수 있었습니다.

팀원분들 모두가 정말 열심히 해 주어 든든하고 의지되어 협업의 감사함을 느꼈고, 막히는 부분은 의견을 구하며 고민하니 혼자서 두 시간 걸리던 부분을 이십 분 만에 할 수 있어 하나의 방법이 안되어도 함께 고민하면 원하는 그림을 구현할 수 있겠다고 느꼈습니다.

# 감사합니다:)