

“ 셀프 인테리어 통합 플랫폼 ”

WIMOS



19101967 강채원

18102002 이현진

Entry procedure.



- 01 Project 개요 및 목표
- 02 Project 진행 방법 및 절차
- 03 Project 설계
- 04 Project 진행 내용
- 05 Project 수행 결과
- 06 설계 요소 평가
- 07 추후 Project 발전 방향
- 08 종합 토의 및 첨부



“ 셀프 인테리어 통합 플랫폼 ”

About WIMOS

셀프 인테리어 통합 플랫폼인 ‘WIMOS’는

“What Is my preference for interior? i make my Mood through Our Space” 의 약자로,

개인별 인테리어 취향에 맞는 정보들을 제공할겠다는 의미와

내가 꾸미는 공간을 통해 나의 기분을 만들겠다는 메시지가 담긴 뜻

01 프로젝트 개요 및 목표

■ 인테리어에 대한 관심 증가와 셀프 인테리어의 호황

- 코로나 19로 '집콕'하는 사람들의 증가와 집값 상승으로 인테리어에 대한 관심도가 크게 증가
- 집값과 물가 상승 등으로 인해 저렴한 가격에 최적의 만족감을 누릴 수 있는 비교 견적 시스템을 활용할 수 있는 인테리어 플랫폼이 인기

■ 다양한 플랫폼의 활용

- 오늘의집, 집닥, 숨고, 한샘, 집꾸미기는 한국의 대표적 인테리어 비교 견적 앱
- 다양한 인테리어 플랫폼들이 호황을 맞이하며 관련 스타트업도 많이 생겨 다양한 플랫폼들을 통한 선택의 폭 증가



여러 플랫폼들을 비교해야 하는 번거로움 ➡ 효과적인 인테리어 비교 분석을 할 수 있는 통합 플랫폼의 필요성

'오늘의 집' 과 '한샘' 두 곳의 가구 정보를 통합하여 개인별 인테리어 취향에 맞는 필터링 결과를 비교하고,
소비자들 간에 인테리어 및 가구 정보에 대한 공유가 가능한 개인별 맞춤 가구 추천 및 커뮤니티 사이트를 만들어보고자 함.

02 Project 진행 방법 및 절차

■ 전반적인 프로젝트 진행 방법

- 팀원 모두 개발 경험이 없는 관계로 익숙하지 않은 개발 협업 툴을 사용하기보다는 팀원 모두에게 익숙한 방식으로 소통하고자 함
- 전반적인 프로젝트 관리 및 공유 : Notion 사용
- code editor : visual studio code를 사용
- code 업로드 및 공유 : github 사용

1. WIMOS 구성 구체화

- 1) 요구 다이어그램을 작성하며 사용자가 WIMOS를 이용하는 과정에서 할 수 있는 행위들을 정의
- 2) 이를 기반으로 MySQL ERD을 작성하며 사용자의 입력 정보를 저장할 테이블과 가구 정보 제공을 위한 테이블의 구성을 논의

2. 가구 정보 크롤링

- 1) 메인페이지에 나타낼 가구 정보 수집 : '한샘'과 '오늘의 집' 가구 정보들을 150여개씩 크롤링하여 furniture Table에 저장
- 2) 개인별 인테리어 취향에 맞는 필터링 기능을 구현하기 위해 가구마다 인테리어 무드에 맞게 분류하는 label을 부여
- 3) label 분류는 모던 시크, 북유럽, 레트로, 로맨틱 네 가지로 해당 정의에 따라 팀원이 직접 부여

02 Project 진행 방법 및 절차

3. WIMOS 페이지별 구성

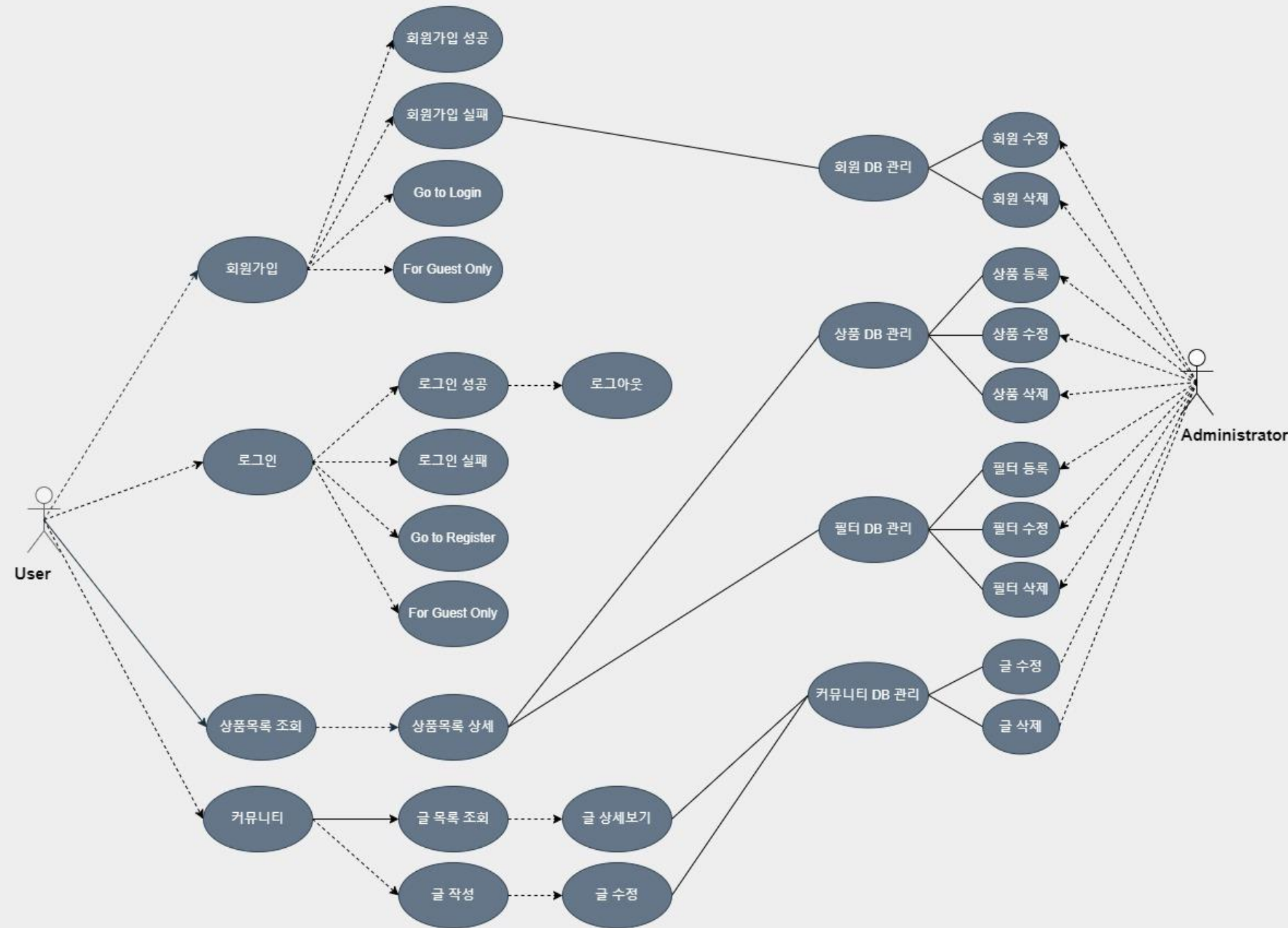
- 1) [furniture] Table 정보를 메인 페이지에 나타낼 수 있도록 구성
 - DB 정보들을 로드할 때마다 새로운 순서로 나타내기 위해 랜덤 정렬 방식 이용
- 2) 로그인, 회원가입, 커뮤니티 페이지 구현
 - 커뮤니티 페이지는 글 작성, 전체 목록 보기, 글 상세 보기, 수정 기능 구현

4. 팀원별 역할

- 강채원
 - WIMOS 구성을 구체화하기 위한 회의 결과를 바탕으로 요구 다이어그램, DB ERD 작성
 - '오늘의 집'과 '한샘' 가구 데이터 크롤링 코드 작성
 - 커뮤니티 페이지 구현 및 최종 발표 PPT를 작성
- 이현진
 - 제공받은 크롤링 코드로 크롤링을 진행한 후, 가구별 label을 부여
 - 메인 페이지와 로그인/회원가입 페이지 구현
 - 기능도 및 FC기능 점수 표와 최종 보고서 작성

03 프로젝트 설계

■ 요구 다이어그램



• 회원가입

- 회원가입 성공 :

신규 유저가 비밀번호 유효성 검사를 모두 통과하여 회원가입 완료된 경우

- 회원가입 실패 :

신규 유저가 비밀번호 유효성 검사를 통과하지 못 했거나 기존 회원인 경우

- Go to Login : 로그인 페이지로 전환

- For Guest Only : 로그인 하지 않은 게스트 모드로 메인 페이지 이동

• 로그인

- 로그인 성공 : 기존 유저가 정확한 회원 정보를 입력한 경우

- 로그아웃 : 로그인 상태인 유저가 다시 미로그인 상태로 돌아감

- 로그인 실패 :

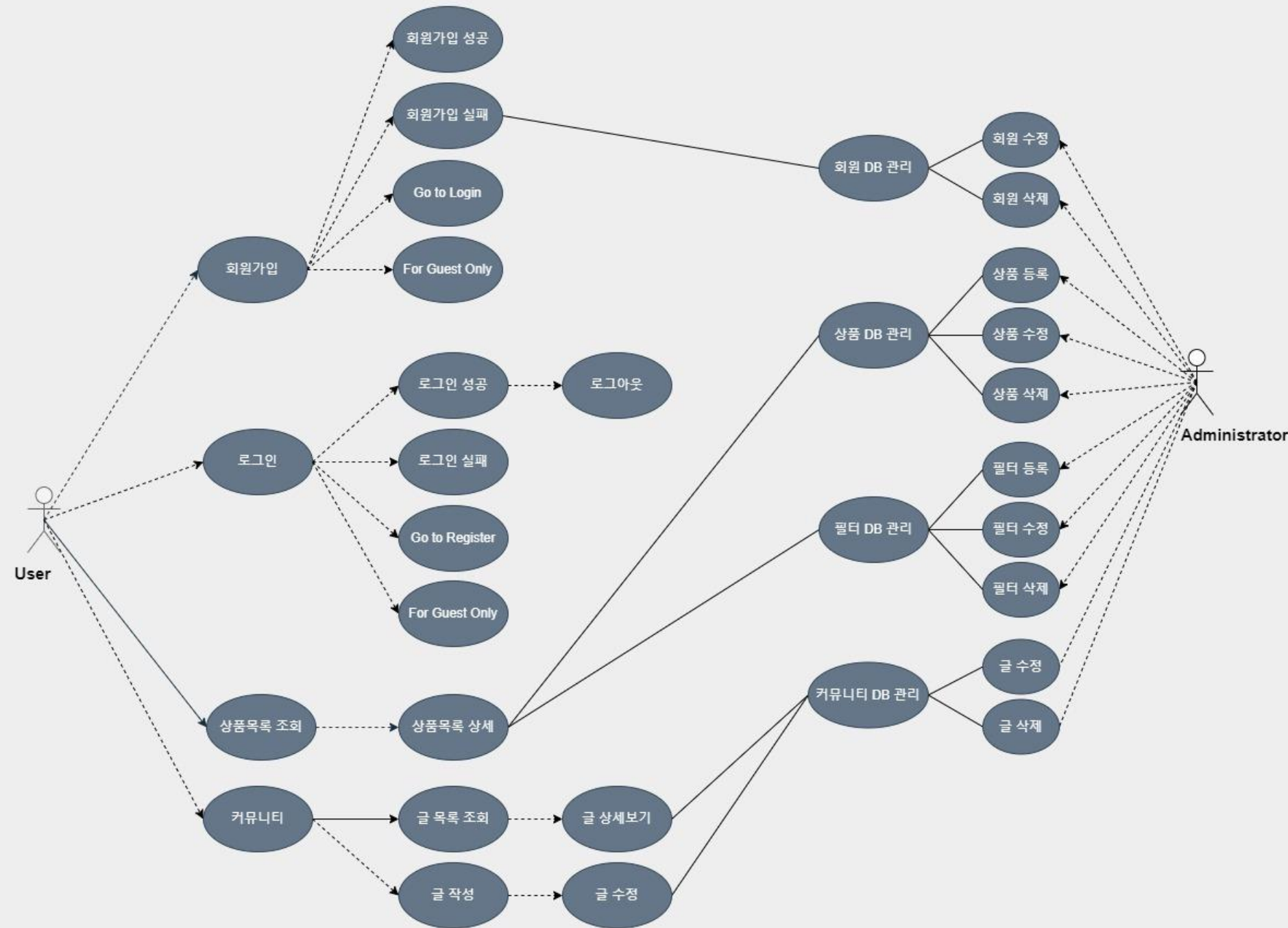
기존 유저가 회원 정보를 잘못 입력한 경우나 기존 유저가 아닌 경우

- For Guest Only : 로그인을 진행하지 않고 Guest 모드로 메인 페이지 이동

- Go to Register : 회원가입 페이지로 이동

03 프로젝트 설계

■ 요구 다이어그램



• 메인 페이지

- 상품 목록 조회 : 기존/신규/게스트 유저는 상품 목록 조회 가능
 - 상품 목록 상세 : 'Detail' 클릭 하여 상품 상세보기 가능

• 커뮤니티 페이지

- 글 목록 조회 :

기존/신규/게스트 유저는 자신 및 다른 사용자들이 작성한 글 목록 확인 가능

- 글 상세보기 : 글 내용 전체를 볼 수 있는 상세 페이지로 이동

- 글 작성 : 기존/신규/게스트 유저는 글 작성 가능

- 글 수정 : 자신이 작성한 글만 수정 가능

• DB 관리

- 사용자의 활동을 통해 입력되는 정보들은 각각 Table로 저장 (사용자, 로그인, 커뮤니티)

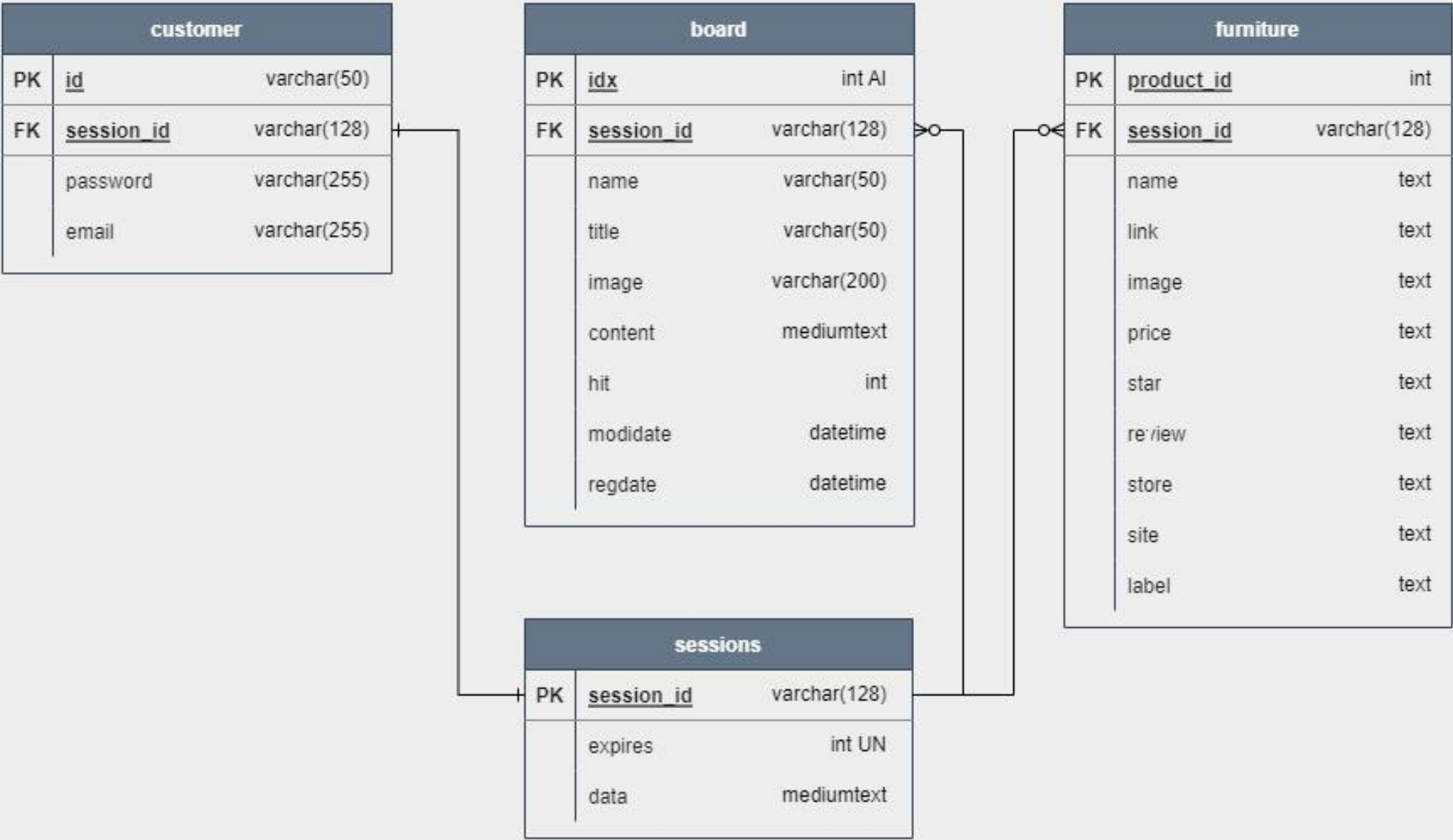
- 메인 페이지에 상품 정보를 나열하기 위해 저장되는 Table

-> 관리자는 DB에서 직접 데이터 추가, 수정, 삭제 작업 통해 관리

03 프로젝트 설계

DB ERD

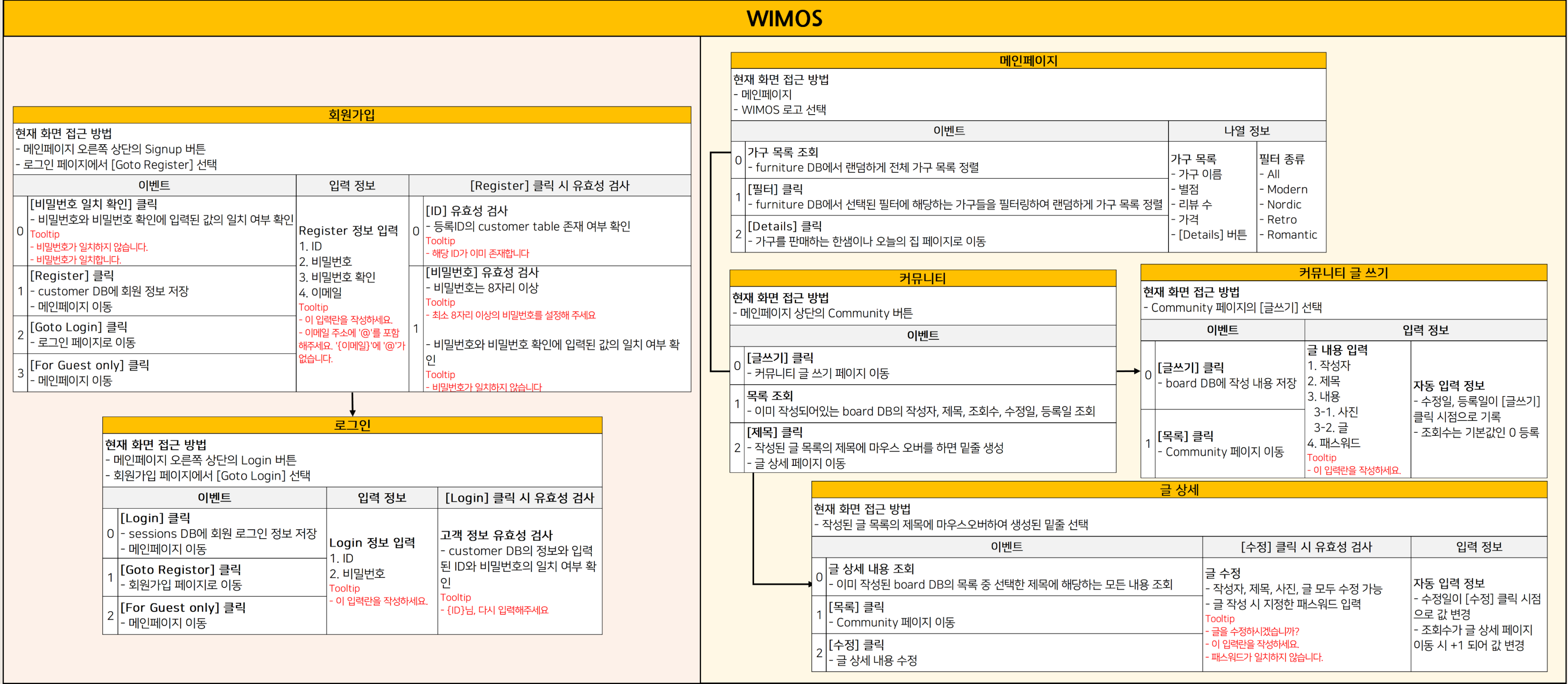
DB ERD



- **Customer**
 - 사용자의 회원가입 데이터 저장 테이블
 - session_id, login_id, login_password, login_email
- **board**
 - 커뮤니티 페이지에서 작성된 글 데이터 저장 테이블
 - session_id, idx, name, title, image, content, hit, modidate, regdate
- **sessions**
 - 사용자의 로그인 데이터 저장 테이블
 - expires, data
- **furniture**
 - 메인 페이지에 나타낼 가구 데이터 저장 테이블
 - product_id, session_id, name, link, image, price, star, review, store, site, label

03 프로젝트 설계

■ 기능도



03 프로젝트 설계

■ 기능도 - Depth 별 기능 정의

번호	Depth1	Depth2	Depth3	기능명
1	GNB	-	-	WIMOS 로고
2		-	-	Home Tab
3		-	-	Community Tab
4		-	-	Login Tab
5		-	-	Signup Tab
6		Community Tab	-	글 상세조회
7			-	글쓰기
8			글쓰기 버튼	글 작성
9				입력란 미작성 alert
10				목록
11				글 수정
12				글 수정 alert
13				패스워드 미입력 alert
14				패스워드 불일치 alert
15		Login	-	Login
16			-	Goto Register
17			-	For Guest Only
18			Login 버튼	입력란 미작성 alert
19				정보 유효성 검사 alert
20		Signup	-	비밀번호 일치 확인
21			-	Register
22			-	Goto Login
23			-	For Guest Only
24			Register 버튼	입력란 미작성 alert
25				이메일 주소 형식 alert
26				ID 유효성 검사 alert
27				비밀번호 조건 alert
28	메인	-	-	All/Modern/ Nordic/Retro/ Romantic
29		-	-	가구 상세 보기

04 Project 진행 내용

주차	진행 내용
10주차	<ul style="list-style-type: none">예제 실습MySQL 서버 연동 테스트크롤링 계획 수립
11주차	<ul style="list-style-type: none">요구 다이어그램 작성DB Schema 설계 및 ERD 작성데이터 크롤링메인 페이지 구축로그인/회원가입 페이지 구성
12주차	<ul style="list-style-type: none">회원가입 페이지 구축로그인 페이지 구축
13주차	<ul style="list-style-type: none">커뮤니티 페이지 구축글 작성 페이지 구축글 상세 페이지 구축
14주차	<ul style="list-style-type: none">글 작성 페이지 구축 수정이벤트 경고창 추가비밀번호 일치 확인 버튼 추가
15주차	<ul style="list-style-type: none">프로젝트 검수기능도 작성요구 다이어그램과 ERD 수정발표 준비

- 10주차 - 프로젝트 시작 단계
 - 계획서를 기반으로 아이디어를 구체화
 - 구현하고자 하는 구성과 비슷한 웹페이지 예제 자료들을 찾아 실습
 - 팀원과 MySQL 서버를 연동, 필요한 테이블 구조와 스키마를 정의
 - 가구 정보 크롤링 계획 수립(python 사용)
- 11주차
 - 요구 다이어그램을 제작하여 사용자 및 관리자의 활동 명세를 작성
 - 구체화된 테이블 구조로 ER Diagram을 작성
 - '오늘의 집'과 '한샘'에서 가구 정보 크롤링 후 필터 라벨링하여 furniture 테이블 구성
 - 네비게이션 바와 메인 페이지를 구성 크롤링한 가구 정보들이 라벨링 필터에 맞게,
매 로드할 때마다 랜덤으로 배열되도록 MySQL 쿼리문으로 불러옴
 - 로그인, 회원가입 페이지는 제공받은 login.html과 register.html을 기반으로 디자인 구성

04 Project 진행 내용

- 12주차 - 회원가입, 로그인 페이지 구현

- 회원가입을 하면 이름, 비밀번호, 이메일 주소 정보가 customer DB에 저장되고, 빈 칸 경고창과 비밀번호 유효성 검사를 실시할 수 있도록 javascript 사용
- 회원가입한 정보를 바탕으로 express 모듈을 이용하여 로그인 시, 회원 여부 파악을 위한 유효성 검사 진행
- 회원 정보가 동일하다면 sessions DB에 'loggedin:true' 의 정보가 저장되고 로그인 되며 네비게이션 바는 Logout 버튼 나타나도록 함
- 로그인한 페이지에서 Logout 버튼을 누르면 sessions DB에서 'loggedin:false' 로 정보를 변환, 네비게이션 바는 원래대로 Login과 Signup 버튼으로 변경

['app.js' - 로그인 결과에 따른 페이지 호출 코드]

[코드 설명]

```
//메인 페이지
app.get('/',(req, res)=>{
  let query = "SELECT * from furniture Order by rand()";
  if(req.session.loggedin == true){
    con.query(query, (err, result)=>{
      err ? res.send(err) : res.render("main_u", {data: result});
    })
  }else{
    let query = "SELECT * from furniture Order by rand()";
    con.query(query, (err, result)=>{
      err ? res.send(err) : res.render("main", {data: result});
    })
  }
});
```

- app.js에서 express 모듈 사용
- 로그인을 성공하고 sessions DB에 loggedin:true의 정보가 저장되면 main_u 페이지를 호출, 그렇지 않다면 main 페이지를 호출하는 코드
- main에는 [Login]과 [Signup] 버튼, main_u에는 [LogOut] 버튼이 구현되어 있음

04 Project 진행 내용

- 13주차 - 커뮤니티 페이지 구현

- Community 탭을 클릭 시, 커뮤니티 페이지로 이동하여 작성된 글들의 목록 조회 가능
- 상세 내용을 보고자 하는 글의 제목에 마우스 오버 시, 밑줄이 생성되며 글 상세 페이지로 이동 가능
- 상세 페이지로 이동 시 해당 글의 조회수(hit)는 1 증가
- 글쓰기 버튼 클릭 후 새 글 작성 및 비밀번호 설정 가능, 이후 설정한 비밀번호를 기반으로 상세 페이지에서 글 수정 가능
- 글 수정 시에는 비밀번호 입력 여부 및 일치 여부를 확인할 수 있는 유효성 검사 진행

['app.js' 조회수 변경 코드]

```
app.get('/read/:idx',function(req,res,next)
{
  var idx = req.params.idx;
  var sql = "select idx, name, title, content, date_format(modidate,'%Y-%m-%d %H:%i:%s') modidate, " +
    "date_format(regdate,'%Y-%m-%d %H:%i:%s') regdate,hit,image from board where idx=?";
  con.query(sql,[idx], function(err,row)
  {
    if(err) console.error(err);
    res.render('read', {title:"글 상세", row:row[0]});
  })
  var hit = req.body.hit;
  data = [idx,hit];
  var sql2 = "update board set hit=hit+1 where idx=?";
  con.query(sql2,data, function(err,result)
  {
    if(err) console.error(err);
  });
});
```

[코드 설명]

- 'app.js'에서 글 상세 페이지를 render 할 때 이미 저장된 hit 정보를 가져와 hit+1을 적용하여 mysql DB를 update하는 코드

04 Project 진행 내용

['app.js' - 상세 페이지 주요 코드]

```
//data 수정
app.post('/update',upload.single("image"),function(req,res,next)
{
    var idx = req.body.idx;
    var name = req.body.name;
    var title = req.body.title;
    var content = req.body.content;
    var passwd = req.body.passwd;
    var datas = [name,title,content,idx,passwd];

    var sql = "update board set name=? , title=?,content=?, modidate=now() where idx=? and passwd=?";
    con.query(sql,datas, function(err,result)
    {
        if(err) console.error(err);
        if(result.affectedRows == 0)
        {
            res.send("<script>alert('패스워드가 일치하지 않습니다.');"history.back();</script>");
        }
        else
        {
            res.redirect('/read/'+idx);
        }
    });
});
```

['read.ejs' - jquery 활용 코드]

```
<script src="https://code.jquery.com/jquery-3.4.1.js"></script>
<script>
    $(document).ready(function(){
        $('#modifybtn').click(function(){
            alert('글을 수정하시겠습니까?');
        });
    });
</script>
```

[코드 설명]

- 'app.js' 에서 커뮤니티 글 상세 페이지에서 데이터 수정을 할 때 입력된 비밀번호의 유효성 검사 코드
- 글 작성 시 설정한 비밀번호와 수정 시 작성한 비밀번호의 일치 여부는 mysql문 활용
- where 절의 idx=?선택된 글을 의미, passwd=?는 수정 시 작성한 비밀번호를 의미
- where절을 만족하는 글이 존재하면 수정,
그렇지 않을 시 "패스워드가 일치하지 않습니다."의 경고창 생성

- jquery 를 활용하여 수정 버튼 클릭 시 "글을 수정하시겠습니까?" 라는 alert창 띄우는 event 생성

04 Project 진행 내용

- 14주차 - 커뮤니티 이미지 파일 추가 기능 구현
 - 커뮤니티 글 작성 시 이미지 파일을 추가할 수 있도록 multer 미들웨어 이용
 - 로그아웃 시 확인하는 경고창과 글 수정 시 수정 확인하는 경고창 추가
 - Ajax 활용하여 회원가입 비밀번호 일치 확인 버튼 구현

['app.js' - multer 모듈 사용 코드]

```
var storage = multer.diskStorage({
  destination: function (req, file, cb){
    cb('', "public/images/");
  },
  filename: function (req, file, cb){
    const ext = path.extname(file.originalname);
    cb('', path.basename(file.originalname, ext) + "-" + Date.now() + ext);
  },
});
var upload = multer({storage:storage});
app.post('/create', upload.single("image"), (req, res)=>{
  const name = req.body.name;
  const title = req.body.title;
  const content = req.body.content;
  const passwd = req.body.passwd;
  const image = `/images/${req.file.filename}`;
  const datas = [name, title, content, passwd, image];
  const sql = `insert into board(name, title, content, regdate, modidate, passwd, hit, image) values(?,?,?,now(),now(),?, 0, ?)`
  con.query(sql, datas)
  res.send("<script>location.href='/community'</script>");
})
```

[코드 설명]

- 'app.js' 파일에서 'multer' 모듈을 사용하여 커뮤니티 글 쓰기 시 사진 파일을 업로드 구현 코드
- 'multer' 는 파일 업로드를 위해 사용되는 Node.js의 미들웨어로 'multipart/form-data' 형식으로 단일 및 다중 파일 업로드를 지원하기 때문에 사용
- public/images/폴더에 업로드 한 파일이 저장되며, 이름 중복을 피하기 위해 기존 이름 뒤에 저장 날짜를 추가하여 저장
- mysql DB의 'board' Table의 'image'에는 '/images/~'로 이미지 경로 데이터 저장

04 Project 진행 내용

['register.ejs' - 패스워드 일치 확인 코드]

[코드 설명]

```
$(document).ready(function(){
    $(".ajaxSend").on("click",function(){
        var pw1 = $('input[name=password]').val();
        var pw2 = $('input[name=password2]').val();
        $.ajax({
            url : "/ajax",
            type : "POST",
            dataType : "JSON",
            data: {"pw1":pw1, 'pw2':pw2}
        })
        .done(function (json){
            if (json.pw1 != json.pw2){
                alert('비밀번호가 일치하지 않습니다. ');
            }else{
                alert('비밀번호가 일치합니다. ')
            }
        })
        .fail(function (xhr, status, errorThrown){
            alert("Ajax Failed")
        })
    })
});
```

- Ajax를 활용하여 \$.ajaxSend 버튼 클릭 시 패스워드 일치 확인 기능 구현 코드
- Method : POST, datatype : json
- pw1 : 처음 작성한 패스워드, pw2 : 일치 확인 위해 두 번째 작성한 패스워드
- 패스워드 불일치 시 '비밀번호가 일치하지 않습니다.' 라는 alert창 띄움
- 패스워드 일치 시 '비밀번호가 일치합니다.' 라는 alert창 띄움
- 데이터 통신 실패 시 'Ajax Failed' 라는 alert창 띄움

04 Project 진행 내용

['read.ejs' - 주요 코드]

```
<tr>
  <th>내용</th>
  <td>
    
    <textarea style="font-family:JejuMyeongjo;font-size:1.2rem;" name="content" id="content" cols="30" rows="10" required><%=row.content%></textarea>
  </td>
</tr>
```

[코드 설명]

- 글 상세 페이지에서 저장된 이미지를 나타낼 때는 위의 'read.ejs'코드와 같이 저장된 경로를 토대로 불러옴

- 15주차 - 프로젝트 최종 검토

- 기능도를 작성


- 요구 다이어그램 및 ERD 수정 및 검토

- 최종 발표 준비 및 시뮬레이션 진행

05 Project 수행 결과

1. 화면 구성

[회원가입 페이지]

 WIMOS

[Home](#) [Community](#)

Register Form

ID

Password

Password

비밀번호 일치 확인


email

Register

Goto Login

For Guest Only

[로그인 페이지]

 [Home](#) [Community](#)

Login Form

[Login](#)

[Goto Register](#)

[For Guest Only](#)

[메인 페이지]

The banner features a dark brown background with the text "WIMOS & MAMMOS" in large white letters. Below it, the tagline "What is my Preference for Interior? i MAKE My Mood through Our Space" is written in a smaller white font. At the top left, there are navigation links for "Home" and "Community". At the top right, there are "Login" and "Signup" buttons. Below the tagline, there are five category filter buttons: "All", "Modern", "Nordic", "Retro", and "Romantic".

[커뮤니티 페이지]


WIMOS

[Home](#)
[Community](#)


[LogOut](#)


우리집 인테리어를 소개해요~

글쓰기

번호	작성자	제목	조회수	수정일	등록일
1	가쁨이	내돈내산 거실 인테리어 구경하고 가세요~><	40	2022-12-06 19:14:15	2022-12-06 19:14:15
2	화분이	식물과 함께하는 우리 집	3	2022-12-06 21:15:10	2022-12-06 21:13:44
3	워니	귀여운 자취방 러그 추천!!	2	2022-12-06 21:17:34	2022-12-06 21:17:34
4	시크쟁이	안녕하세요 시크충입니당 ㅎㅎㅎ	3	2022-12-06 21:18:48	2022-12-06 21:18:48
5	공통등인사	색감 예쁜 스물 추천해용	3	2022-12-06 21:20:27	2022-12-06 21:19:15
6	tomntoms	복유럽 감성의 침대 너무 예뻐서 추천!!!	7	2022-12-06 22:05:35	2022-12-06 21:21:58

[글 쓰기 페이지]

 WIMOS

[Home](#) [Community](#)

[LogOut](#)


커뮤니티 글 쓰기

작성자	
제목	
내용	<div><div>파일 선택</div><div>선택된 파일 없음</div></div>
패스워드	

목록

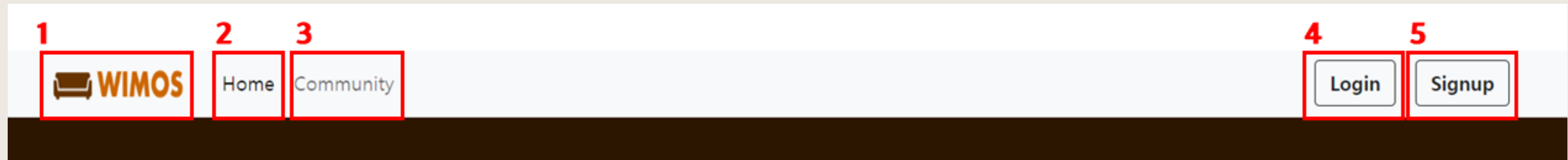
글쓰기

[글 상세 및 수정 페이지]

작성자	기쁨이
제목	내돈내산 거실 인테리어 구경하고 가세요~><
내용	 <p>밝고 화사한 톤의 인테리어를 해봤어요~ 이건 일부분에 불과하답니다. 더 많은 제 인테리어 작품 보고 싶으시면 제 블로그 놀러오세요~!</p>

05 Project 수행 결과

■ 2. GNB(네비게이션 바)



- 1&2 : 'WIMOS' 로고(1) 또는 'Home'(2) 클릭 시 메인 페이지로 이동
- 3 : 'Community'(3) 클릭 시 커뮤니티 페이지로 이동
- 4 : 'Login'(4) 클릭 시 로그인 페이지로 이동
- 5 : 'Signup'(5) 클릭 시 회원가입 페이지로 이동

05 Project 수행 결과

3. 회원가입 페이지

The screenshot shows the WIMOS registration form. At the top left, the WIMOS logo and navigation links 'Home' and 'Community' are enclosed in a red box labeled '1'. The form itself is titled 'Register Form'. It contains several input fields: 'ID', 'Password' (twice), and 'email'. A button labeled '비밀번호 일치 확인' (Check Password Match) is located between the two password fields. Below the input fields are three large buttons: 'Register' (labeled '4'), 'Goto Login' (labeled '5'), and 'For Guest Only' (labeled '6'). A red box labeled '2' encompasses the entire registration form area.

Three stacked alert boxes showing validation results. Each box has a title 'localhost:3000 내용:' and a message, followed by a blue '확인' (Confirm) button.

- Box 1 (green border): 내용: 해당 ID가 이미 존재합니다.
- Box 2 (purple border): 내용: 비밀번호가 일치하지 않습니다.
- Box 3 (green border): 내용: 비밀번호가 일치합니다.

Below these boxes is a yellow-bordered box containing an email input field and a message: '이 입력란을 작성하세요.' (Please enter in this field).

- 1 : 네비게이션바(1)와 같은 기능
- 2,4 : 회원정보(2)를 입력하고 Register(4)를 클릭하면 유효성 검사를 진행한 후 회원가입을 진행

유효성 검사 결과 해당 ID가 이미 customer DB에 이미 존재/ 비밀번호가 8자리 미만/ 비밀번호 불일치 시, 위 그림의 초록색 박스와 같은 경고창 생성

05 Project 수행 결과

3. 회원가입 페이지

The screenshot shows the WIMOS registration form. The header includes the WIMOS logo and navigation links for Home and Community. The form is titled 'Register Form' and contains several input fields and buttons. Red boxes and numbers 1 through 6 highlight specific elements: 1 points to the header, 2 points to the ID input field, 3 points to the password input fields, 4 points to the '비밀번호 일치 확인' (Check password match) button, 5 points to the 'Register' button, 6 points to the 'Goto Login' button, and 7 points to the 'For Guest Only' button.

Three screenshots of the WIMOS registration form showing different validation messages. The first screenshot shows a message: 'localhost:3000 내용: 해당 ID가 이미 존재합니다' (localhost:3000 content: The corresponding ID already exists). The second screenshot shows a message: 'localhost:3000 내용: 비밀번호가 일치하지 않습니다.' (localhost:3000 content: Passwords do not match). The third screenshot shows a message: 'localhost:3000 내용: 비밀번호가 일치합니다.' (localhost:3000 content: Passwords match). Each message is followed by a blue '확인' (Confirm) button.

- 3 : 두 번의 비밀번호 입력 후 [비밀번호 일치 확인] 버튼(3) 클릭 시, 경고창과 함께 일치 여부 확인 가능

공백 칸 존재/올바르지 않은 이메일 형식 -> 노란색 박스와 같이 주의 메시지 생성

- 5 : [Goto Login] 버튼(5) 클릭 시 로그인 페이지로 이동
- 6 : [For Guest Only] 버튼(6) 클릭 시 메인 페이지로 이동

05 Project 수행 결과

■ 4. 로그인 페이지

The screenshot shows the WIMOS login page. At the top left, the WIMOS logo and navigation links 'Home' and 'Community' are enclosed in a red box labeled '1'. In the center, a 'Login Form' contains an 'ID' input field (labeled '2'), a 'Password' input field, a 'Login' button (labeled '3'), a 'Goto Register' button (labeled '5'), and a 'For Guest Only' button (labeled '6'). All these elements are outlined with red boxes.

The top screenshot, enclosed in a green box, shows a message box with the text 'localhost:3000 내용: 이현진님, 다시 입력해주세요' and a blue '확인' button. The bottom screenshot, enclosed in a yellow box, shows the login form with a red error message '이 입력란을 작성하세요.' next to the password field.

- 1 : 네비게이션바(1)의 기능과 동일
- 2,3 : 회원정보(2) 입력 후 Login(3)을 클릭 시, 유효성 검사를 진행한 후 로그인 가능

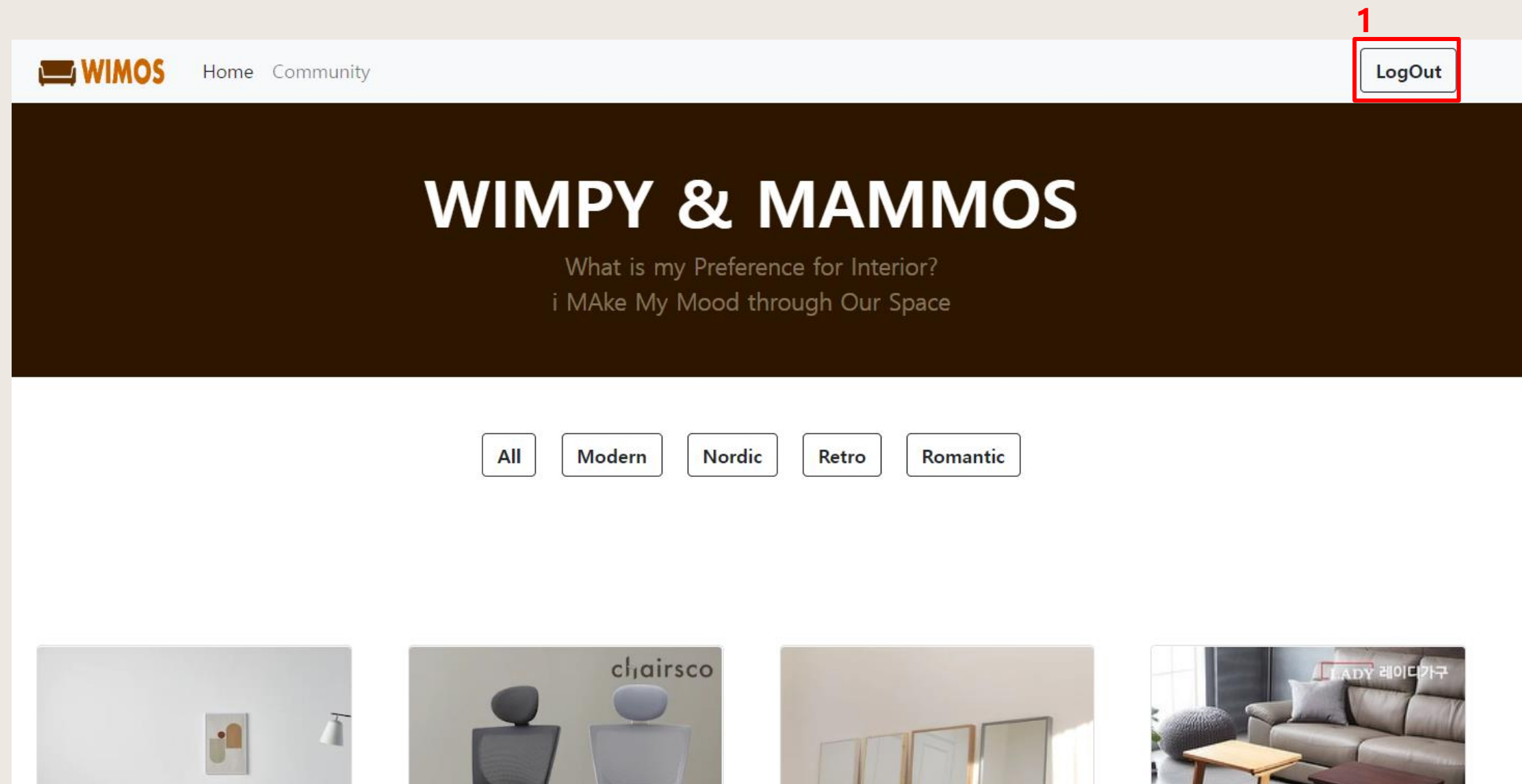
유효성 검사 결과 ID나 비밀번호 중 하나라도 customer DB에 저장된 내용과 불일치 시 위 그림의 초록색 박스와 같은 경고창 생성

공백칸 존재 시 위 그림의 노란색 박스와 같이 주의 메시지 생성

- 4,5 : [Goto Register] 버튼(4) 클릭 시 로그인 페이지로 이동, [For Guest Only] 버튼(5) 클릭 시 메인 페이지로 이동

05 Project 수행 결과

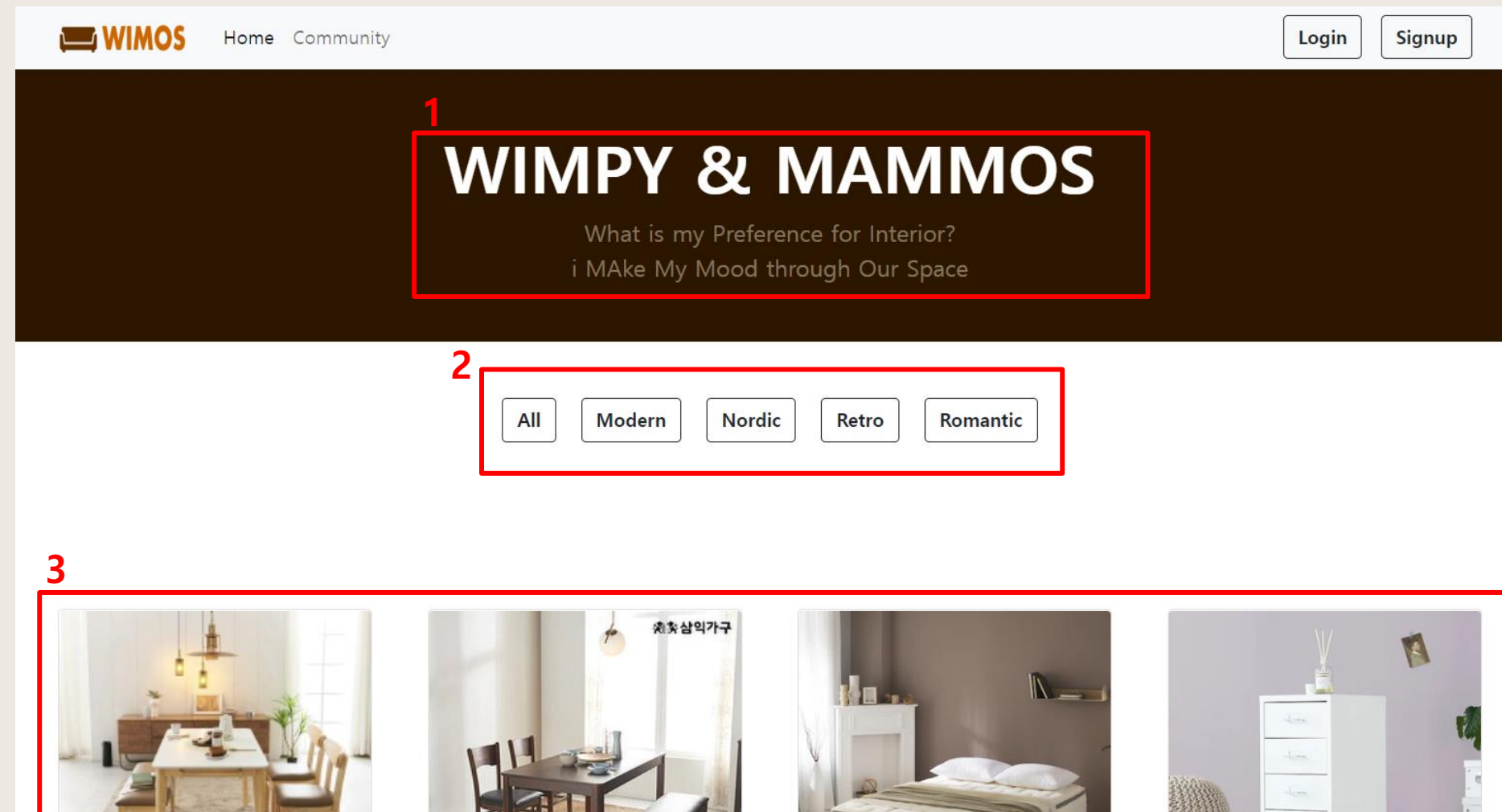
■ 4. 로그인 페이지



- 로그인에 성공 시 메인 페이지로 돌아가고, 네비게이션 바는 'LogOut'(1)으로 변경

05 Project 수행 결과

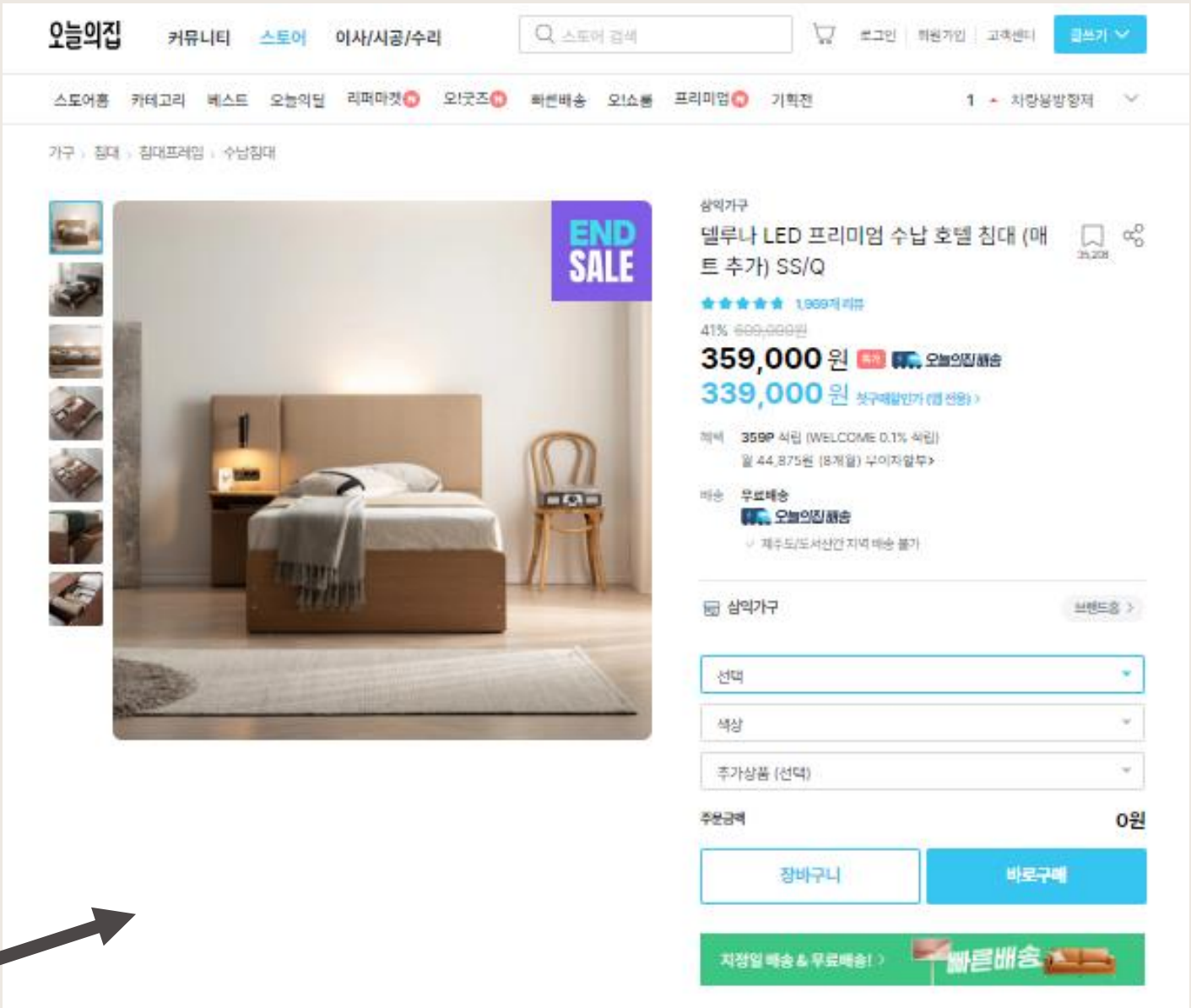
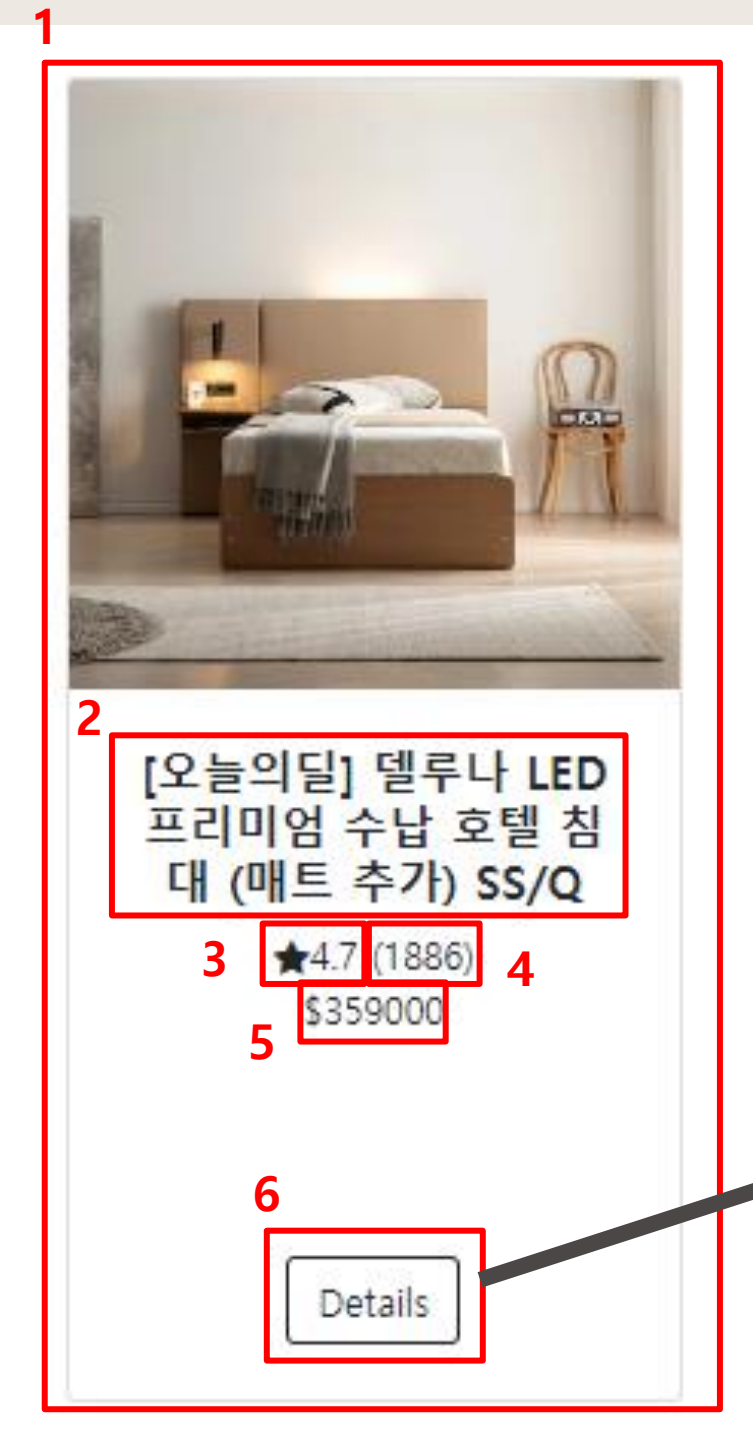
■ 5. 메인 페이지



- 1 : 사이트 명인 WIMOS의 의미를 담은 문구(1)
- 2 : All(전체), Modern(모던시크), Nordic(북유럽 풍), Retro(레트로), Romantic(로맨틱) 필터(2)가 가구 목록 가장 위에 나열되며
필터에 따라 가구 목록 확인 가능
- 3 : 필터에 따른 가구 목록(3) 확인 가능

05 Project 수행 결과

5. 메인 페이지 - 가구 목록




[6 클릭 시 연결된 링크로 이동]

- 1,2,3,4,5,6 : 가구 목록에서는 (1)과 같이 제목(2), 별점(3), 리뷰 수(4), 가격(5) 확인 가능
- 6 : [Details] 버튼(6) 클릭 시 저장되어 있는 사이트 URL로 접근 가능

05 Project 수행 결과

■ 6. 커뮤니티 페이지

 Home Community

LogOut

2

00우리집 인테리어를 소개해요~

4

글쓰기

번호	작성자	3 제목	조회수	수정일	등록일
1	기쁨이	내돈내산 거실 인테리어 구경하고 가세용~~><	39	2022-12-06 19:14:15	2022-12-06 19:14:15
2	화분이	식물과 함께하는 우리 집	3	2022-12-06 21:15:10	2022-12-06 21:13:44
3	워니	귀여운 자취방 러그 추천!!	2	2022-12-06 21:17:34	2022-12-06 21:17:34
4	시크쟁이	안녕하세요 시크충입니당 ㅎㅎㅎㅎ	3	2022-12-06 21:18:48	2022-12-06 21:18:48
5	공릉동인싸	색감 예쁜 스톨 추천해용	3	2022-12-06 21:20:27	2022-12-06 21:19:15
6	tomntoms	북유럽 감성의 침대 너무 예뻐서 추천!!!	7	2022-12-06 22:05:35	2022-12-06 21:21:58

- 1 : 네비게이션바(1)의 기능과 동일
- 2 : 커뮤니티 페이지는 이전에 작성된 글 목록(2) 확인 가능
- 3 : 제목(3)에 마우스 오버 시 밑줄 생성 및 클릭 시 글 상세 페이지로 이동
- 4 : 글쓰기(4) 버튼 클릭 시 글을 직접 작성할 수 있는 페이지로 이동

05 Project 수행 결과

■ 7. 커뮤니티 글 쓰기

WIMOS Home Community LogOut

커뮤니티 글 쓰기

작성자	
제목	
내용	<div>파일 선택 선택된 파일 없음</div> <div></div>
패스워드	

목록 글쓰기

- 1,2 : 네비게이션바(1)의 기능과 동일함. 작성자, 제목, 사진 파일 선택, 글, 패스워드를 작성하여 정보 입력(2)
- 3 : 목록(3) 버튼 클릭 시 커뮤니티 페이지로 이동
- 4 : 글쓰기(4) 버튼 클릭 시 빈 칸이 있을 때에는 칸을 채워 달라는 주의 메시지가 나타나도록 하고,


그렇지 않을 시 board Table에 데이터가 추가 및 커뮤니티 페이지로 이동

글 수정일과 등록일이 자동으로 글쓰기 버튼(4)을 선택한 시간으로 입력되고 조회수는 0으로 입력됨

05 Project 수행 결과

8. 커뮤니티 글 상세보기 및 수정

글 상세

작성자	기쁨이
제목	내돈내산 거실 인테리어 구경하고 가세요~>><
내용	<div></div> <p>밝고 화사한 톤의 인테리어를 해봤어요~ 이건 일부분에 불과하답니다. 더 많은 제 인테리어 작품 보고 싶으시면 제 블로그 놀러오세요~! 함께 소통해요!! https://blog.naver.com/riQmadl/223333 이웃추가 늘 환영합니다~~~</p>
1 패스워드	
변경일	2022-12-06 19:14:15
등록일	2022-12-06 19:14:15
조회수	39

23
목록수정

localhost:3000 내용:
글을 수정하시겠습니까?
확인

패스워드	
변경일	이 일련번호를 작성하세요.

localhost:3000 내용:
패스워드가 일치하지 않습니다.
확인

- 1 : 패스워드 입력 칸, 글 작성자만 수정할 수 있는 권한을 부여하기 위한 설정으로 글 수정 시 필수 사항임
- 2 : 목록(2) 버튼 클릭 시 커뮤니티 페이지로 이동
- 3 : 수정(3) 버튼을 누르면 위 그림의 초록색 박스와 같이 경고창이 생성, 패스워드 미 입력 시 위 그림의 노란색 박스와 같은 주의 메시지 생성

패스워드 불일치 시 위그림의 보라색 박스와 같은 경고창 생성 및 글 수정 실패

글 수정 성공 시, 그 시점에 따라 변경일 데이터가 변경됨

05 Project 수행 결과

■ 9. 로그아웃

localhost:3000 내용:
로그아웃 하시겠습니까?

확인



Home Community

Login

Signup

- 로그아웃 시 “로그아웃을 하시겠습니까?”를 묻는 경고창이 생성됨
- 확인 클릭 시 로그인 전 화면인 네비게이션 바가 [Login], [Signup] 버튼으로 구성된 페이지로 이동

06 설계 요소

■ 성능

- 데이터의 구현에 관해 높은 성능은 null값을 허용하지 않거나 데이터 데이터의 형식을 맞추는 것들을 의미한다고 할 때
- 고객 데이터를 기반으로 로그인과 로그아웃을 다루는 세션 데이터, 가구 정보들을 나열하여 메인 페이지에 나타내기 위한 가구 데이터, 커뮤니티 글 추가 및 수정을 위한 커뮤니티 데이터들이 모두 안정적으로 구현되었다는 점에서 우수한 성능을 가짐
- 기능 구현에 관해 높은 성능은 새로운 데이터 값이 입력되고 수정되는 것에 대한 오류 없이 원활하게 진행되는 것
- 이 점에서 로그인, 가구 정보 제공, 커뮤니티 글 작성 및 수정 모두 설계 요소에 맞는 올바른 구현이 되어 있어 우수한 성능을 가짐

■ 안전성

- 본 프로젝트의 웹페이지 중 메인 페이지에 가구정보를 나타낼 때 구현해둔 local DB 에서 데이터 가져옴
-> 해킹 위험 적다는 장점
- 현재 가구 상세 정보를 확인 위해 버튼을 클릭 시 DB에 저장된 링크인 외부 페이지로 연결되어 나타나게 구현
-> 이는 원본 사이트의 해킹/오작동 등 그 영향을 고스란히 받을 수 있다는 위험과 가짜 클릭 버튼을 만들어내는 해킹이 일어날 위험 존재
-> 이를 해결하기 위해 자체 사이트 구성이나 AWS와 같은 서버 활용으로 가구 상세 페이지를 구성하여 안전성을 향상시킬 수 있을 것으로 보임

06 설계 요소

■ 보안성

- 커뮤니티 페이지의 글 수정 기능은 유저가 글을 작성할 때 설정한 비밀번호를 똑같이 입력해야만 가능하도록 설정

-> 이는 유저 간의 보안을 강화한 작업

-> 추후 글 작성 비밀번호를 로그인 패스워드와 연동하여 유저 입장에서의 편리성 증가 및 보안성 강화 가능

- 현재 : 회원가입 패스워드 정보가 현재는 암호화되지 않은 채 customer DB에 저장

- 문제점 : 유저가 비밀번호를 잊어버리는 등의 문제가 발생하였을 때 조치할 수 있는 방법이 관리자가 직접 비밀번호를 제공하는 방법 뿐

-> 회원가입 시 등록한 비밀번호를 암호화 처리를 거쳐서 저장되는 것으로 발전시켜 사용자 정보의 보안성 증대 및 개인정보 유출에 대비 가능

-> 또한, 사용자 인증 절차 구현 통해 관리자의 개입 없이 확실한 신원 보장의 절차 마련 가능

06 설계 요소

■ 구현 기간

- 본 프로젝트는 1주 간의 프로젝트 계획 수립, 이후 4주 간 로그인/회원가입 페이지, 메인 페이지, 커뮤니티 페이지 구현, 1주 간의 프로젝트 검토 및 발표 준비가 이뤄짐
- 웹 페이지 보완 및 을 고려하여 해당 프로젝트 구현을 위해서 총 6주의 기간이 필요하다고 예상
- 현재는 통합 쇼핑몰 아이디어를 제안할 수 있는 기반을 다지는 페이지로 구현이 되어있으나, 추가 발전 사항 구현을 위해서는 더 긴 구현 기간이 필요

■ 개인의 로드

- 현재 : 로컬 단일 서버로 구축되어 있으며, 소수의 사용자가 WIMOS를 사용하는 데에는 구현 문제 없음
 - 하지만 웹페이지가 상용화되고 사용자가 폭등하게 될 EO를 대비한 서버 구축이 필요
- > 사용자의 폭등을 감당할 수 있도록 분산 서버 시스템을 활용하거나 AWS 등의 안정성이 보장된 서버를 사용
- > 사용자 수에 따른 제약으로부터 자유롭게 할 수 있으며, 이는 개인의 로드를 원활하게 하는 데에 도움이 됩니다.

06 설계 요소 평가

■ 구현 단가 - FC 기능 점수 사용

개발대상 SW 기능점수 산정 (FP 정통법)

데이터/트랜잭션 기능	기능타입	복잡도	FP값 (미 조정값)
고객 데이터	ILF	보통	10
세션 데이터	ILF	낮음	7
가구 데이터	ILF	낮음	7
커뮤니티 데이터	ILF	낮음	7
고객 데이터 추가	EI	낮음	3
고객 데이터 조회	EQ	낮음	3
세션 데이터 추가	EI	낮음	3
세션 데이터 조회	EQ	낮음	3
세션 데이터 수정	EI	보통	4
세션 데이터 삭제	EI	낮음	3
가구 데이터 출력	EO	낮음	4
커뮤니티 데이터 추가	EI	낮음	3
커뮤니티 데이터 조회	EQ	낮음	3
커뮤니티 데이터 출력	EO	낮음	4
커뮤니티 데이터 수정	EI	보통	4
FP 계산 값			68



보정 전 개발원가 산정

총 기능점수	기능점수 단가	보전전개발원가
68	553,114	37,611,752

구분	적용기준	보정계수
개발규모	$0.4057 \times (\log_e(\text{기능점수}) - 7.1978)^2 + 0.8878$	1.28
연계복잡성	2개의 타기관 연계	0.94
성능 요구수준	피크타임에 중요함	1.00
운영환경 호환성	요구사항 없음	0.94
보안성	암호화, 개인정보보호	1.00

- SW사업 대가산정 가이드에 따라 보정계수 산정

06 설계 요소 평가 - 6. FC 기능 점수

보정 후 개발원가 산정

총 기능	기능 점수당	보정계수					금액(원)
		규모	연계	성능	운영환경	보안성	
68	553,114	1.28	0.94	1.00	0.94	1.00	42,539,192

직접경비 및 이윤 산정

산정기준	개발원가	적용률	금액(원)
개발원가의 25%이내	42,539,192	25%	10,634,798

구분	산출내역	금액(원)
보고서인쇄비	상세요구분석서 등 4종 X 4부 (50원 X 1,000매 X 4부 X 4종 = 800,000원)	800,000
출장여비	2인 X 2회(2박3일 기준) (200,000원 X 2인 X 2회 = 800,000원)	800,000
합계		1,600,000

소프트웨어 개발비 산정

총 기능	기능 점수당	보정계수					금액(원)
		규모	연계	성능	운영환경	보안성	
68	553,114	1.28	0.94	1.00	0.94	1.00	42,539,192
이윤 (25)%							10,634,798
직접경비							1,600,000
소프트웨어 개발비(부가세 별도)							54,773,990
소프트웨어 개발비 (부가세 포함)							60,251,389

07 추후 Project 발전 방향

■ 마이페이지 구현 및 회원 정보 추가

활용

- 사용자 정보를 이용하여 사용자 정보가 한눈에 보일 수 있도록 마이페이지 구성.
 - 특히 가구들을 제공하는 메인 페이지에서 좋아요 기능을 구현하여 이를 마이페이지에서 사용자가 한눈에 볼 수 있도록 함.
 - 커뮤니티 페이지에서 글을 수정 시 사용자가 매번 글을 작성할 때마다 설정한 비밀번호를 기억해야하는 번거로움
- > 회원이 글 작성 시 회원 가입 정보인 패스워드로 글 작성 비밀번호가 설정되도록 하여 회원은 별도의 비밀번호 작성 없이 글 수정 가능하도록 함.
- > 비회원인 경우에만 직접 비밀번호를 설정하도록 기능을 변경, 이에 따라 회원이 작성한 글과 비회원이 작성한 글 구분 가능

■ 글 삭제 & 댓글 기능

추가

- 글 삭제와 댓글 기능을 추가하여 양방향 소통이 가능하도록 구성

■ 필터

추가

- 현재 적용 중인 필터에 무드 뿐만 아니라 색감, 재질, 크기, 등 다양한 종류의 필터 추가
- 크롤링한 데이터의 라벨을 다양하게 세분화하여 구현 가능

07 추후 Project 발전 방향

■ 회원별 맞춤 추천 시스템

- 사용자가 직접 필터링 항목을 선택하지 않아도 개인화 시스템을 적용하여 회원가입 시 파악한 회원의 취향을 고려한 자동 추천 시스템도 구현 가능

-> 개인화 마케팅 트렌드를 반영한 웹 페이지

- 이에 따라 팀원이 직접 부여하는 label에서도 변화를 줌

■ 머신러닝 & 딥러닝 적용

- 방대한 양의 정보를 크롤링하여 라벨링한 결과를 바탕으로 머신러닝과 딥러닝을 활용하여 라벨링 모델 개발

-> 새롭게 제공받는 데이터 자동 라벨링 시스템 구축

- 현재 : 크롤링한 저장된 정보를 페이지에 띄워주는 정적인 방법

-> 실시간 크롤링으로 자동화하여 동적으로 제공 데이터를 변경 가능

07 추후 Project 발전 방향

■ 자체 사이트 구현

- 현재 : 상품의 상세 정보 클릭 시 크롤링한 외부 사이트로 접속하여 제품 구매 가능

-> 이를 자체 시스템으로 변환하여 상품 상세 정보를 진행하고 결제 및 장바구니 시스템까지 추가하여 더욱 유의미한 웹페이지로 발전

-> 결제를 기반으로 한 사용자 구매 데이터도 생성할 수 있어 마케팅 방안에도 활용 가능

-> 제휴 및 할인 쿠폰이나 이벤트 등 자체적으로 진행 가능한 시스템으로 확장 가능

08 종합 토의 및 첨부

■ 통합 플랫폼의 역할 및 상용화 가능성

- 한 가지 도메인으로 분산되어있는 다양한 플랫폼들을 통합하고자 했다는 점에서 의의
- '오늘의 집', '한샘' 등 온라인 쇼핑물들을 통합한 플랫폼이 많지만, 통합 플랫폼들이 이용자 입장에서 쇼핑을 하는 데에 복잡도가 증가
- 그 초기 단계의 웹페이지를 구현하고자 계획한 부분을 잘 수행하고 정확한 역할 분담이 되어 효율적으로 프로젝트를 수행했다는 점에서도 긍정적
- 추후 프로젝트 발전 사항들을 잘 적용하고 온라인 쇼핑 사이트의 형태를 안정성있게 갖추게 된다면, 상용화되어 활용되기 충분

■ 개인별 맞춤 추천 시스템

- 추후 회원 정보를 활용한 마이페이지 구성 및 개인화 추천 시스템 기능을 구현 시 유저에게 더욱 편리한 웹페이지로 발전 가능
- 가구, 인테리어에 국한되지 않고 다양한 도메인에 적용 가능 아이디어
- 온라인 쇼핑이 증가하고, 창업이 증가하고 있는 트렌드에 적합한 프로젝트

08 종합 토의 및 첨부

■ 첨부

- 첨부 1. Project 계획서
- 첨부 2. 주차별 보고서
- 첨부 3. 발표 자료
- 첨부 4. 최종 보고서
- 첨부 5. 소스코드
- '오늘의 집', '한샘' 링크

감사합니다

