

HOCAR PROJECT

Member : 김동진, 김지환, 소강석, 민경환 , 김주원

INDEX

01

DB 구조

03

시현

02

코드

사용자
관리자

04

자체 피드백

DB 구조

Table: **user**

Columns:

name	varchar(20)
<u>id</u>	varchar(20) PK
pw	varchar(20)
phone_num	varchar(30)
address	varchar(50)
email	varchar(20)
email2	varchar(20)

Table: **usedcar**

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int
fuel_type	varchar(20)
color	varchar(10)

Table: **wishbuylist**

Columns:

wishBuyUser_name	varchar(20)
id	varchar(20)
phone_num	varchar(20)
car_number	varchar(30)
email	varchar(20)
email2	varchar(20)

Table: **wishselllist**

Columns:

wishSellUser_name	varchar(20)
id	varchar(20)
phone_num	varchar(20)
car_number	varchar(30)
email	varchar(20)
email2	varchar(20)

DB 구조

Table: **user**

Columns:

name	varchar(20)
<u>id</u>	varchar(20) PK
pw	varchar(20)
phone_num	varchar(30)
address	varchar(50)
email	varchar(20)
email2	varchar(20)

Table: **usedcar**

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int
fuel_type	varchar(20)
color	varchar(10)

Table: **wishbuylist**

Columns:

wishBuyUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
car_number	varchar(30)
email	varchar(20)
email2	varchar(20)

Table: **wishselllist**

Columns:

wishSellUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
car_number	varchar(30)
email	varchar(20)
email2	varchar(20)

DB 구조

Table: **user**

Columns:

name	varchar(20)
<u>id</u>	varchar(20) PK
pw	varchar(20)
phone_num	varchar(30)
address	varchar(50)
email	varchar(20)
email2	varchar(20)

Table: **usedcar**

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int
fuel_type	varchar(20)
color	varchar(10)

Table: **wishbuylist**

Columns:

wishBuyUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
<u>car_number</u>	varchar(30)
email	varchar(20)
email2	varchar(20)

Table: **wishselllist**

Columns:

wishSellUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
<u>car_number</u>	varchar(30)
email	varchar(20)
email2	varchar(20)

DB 구조

Table: user

Columns:

name	varchar(20)
<u>id</u>	varchar(20) PK
pw	varchar(20)
phone_num	varchar(30)
address	varchar(50)
email	varchar(20)
email2	varchar(20)

Table: usedcar

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int

Table: searchlist

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int
fuel_type	varchar(20)
color	varchar(10)

Table: wishbuylist

Columns:

wishBuyUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
<u>car_number</u>	varchar(30)
email	varchar(20)
email2	varchar(20)

Table: wishselllist

Columns:

wishSellUser_name	varchar(20)
<u>id</u>	varchar(20)
phone_num	varchar(20)
<u>car_number</u>	varchar(30)
email	varchar(20)
email2	varchar(20)

코드

```
<form method="get" action="Board_Buy_Search.jsp">
  <h4>차종</h4>
  <input type="checkbox" name="type" value="경차"> 경차
  <br>
  <input type="checkbox" name="type" value="소형">
소형
  <br>
  <input type="checkbox" name="type" value="준중형">
준중형
  <br>
  <input type="checkbox" name="type" value="중형">
중형
  <br>
  <input type="checkbox" name="type" value="대형">
대형
  <br>
  <input type="checkbox" name="type" value="SUV">
SUV
  <br>
```

차종

- ☐ 경차
- ☐ 소형
- ☐ 준중형
- ☐ 중형
- ☐ 대형
- ☐ SUV

Board_Buy_Search.jsp

Board_Buy_List.jsp

Board_Buy.jsp

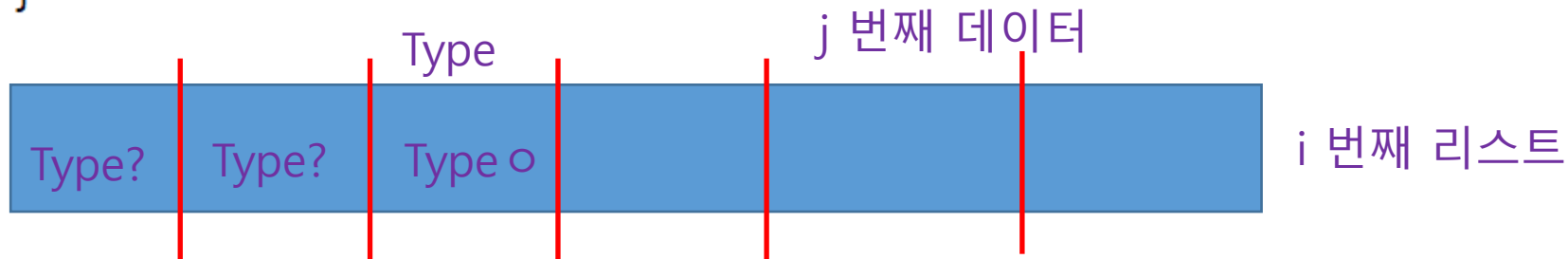
```
61 table, th, td {
62     border: 1px solid black;
63     border-collapse: collapse;
64 }
65 </style>
66 </head>
67 <body>
68
69 <%
70 String [] type = request.getParameterValues("type");
71 String [] company = request.getParameterValues("company");
72 String [] fuel = request.getParameterValues("fuel");
73 String [] color = request.getParameterValues("color");
74
```

코드

```
78 ArrayList<usedCarVo> searchCarList = new ArrayList<usedCarVo>();
79 ArrayList<usedCarVo> bucket1 = new ArrayList<usedCarVo>();
80 ArrayList<usedCarVo> bucket2 = new ArrayList<usedCarVo>();
81 ArrayList<usedCarVo> bucket3 = new ArrayList<usedCarVo>();
82 ArrayList<usedCarVo> bucket4 = new ArrayList<usedCarVo>();
83
84 searchCarList = sc.searchCar();
85
86 if(type == null){
87     for(int i=0; i<searchCarList.size() ; i++){
88         bucket1.add(searchCarList.get(i));
89     }
90 } else if(type != null) {
91     for(int i=0; i<searchCarList.size() ; i++){
92         for(int j=0; j < type.length; j++){
93             if(searchCarList.get(i).getcar_type().equals(type[j])){
94                 bucket1.add(searchCarList.get(i));
95             }
96         }
97     }
98 }
```

검색 결과들을 담을 중간 바구니 생성

전체 DB 데이터 입력 받음



시현

```
100 if(company == null){
101     for(int i=0; i<bucket1.size() ; i++){
102         bucket2.add(bucket1.get(i));
103     }
104 } else if(company != null) {
105     for(int i=0; i<bucket1.size() ; i++){
106         for(int j=0; j < company.length; j++){
107             if(bucket1.get(i).getmade_company().equals(company[j])){
108                 bucket2.add(bucket1.get(i));
109             }
110         }
111     }
112 }
113
114 if(fuel == null){
115     for(int i=0; i<bucket2.size() ; i++){
116         bucket3.add(bucket2.get(i));
117     }
118 } else if(fuel != null) {
119     for(int i=0; i<bucket2.size() ; i++){
120         for(int j=0; j < fuel.length; j++){
121             if(bucket2.get(i).getfuel_type().equals(fuel[j])){
122                 bucket3.add(bucket2.get(i));
123             }
124         }
125     }
```

N개의 칼럼, 항목들에
대해 반복 시행

시현

```
for(int i = 0; i < bucket4.size(); i++){  
    sc.insertSearchList(bucket4.get(i).getcar_number(), bucket4.get(i).getcar_ty  
        bucket4.get(i).getprice(), bucket4.get(i).getmade_year(), bucket4.ge  
}
```

복사본 테이블에 값 저장

Table: **searchlist**

Columns:

<u>car_number</u>	varchar(30) PK
car_type	varchar(20)
car_name	varchar(60)
made_company	varchar(20)
price	int
made_year	int
dis_driven	int
fuel_type	varchar(20)
color	varchar(10)

<%

```
int pagePerRow1 = 30;
```

```
String listSql = "SELECT * FROM SEARCHLIST ORDER BY CAR_TYPE DESC LIMIT ?, ?";
```

```
listStatement = connection.prepareStatement(listSql);  
listStatement.setInt(1, (currentPage1 - 1) * pagePerRow1);  
listStatement.setInt(2, pagePerRow1);  
listResultSet = listStatement.executeQuery();
```

%>

복사본 테이블에 저장된 값 출력

시현

```
String id=request.getParameter("id");
String pw=request.getParameter("pw");

 userDao udao=new userDao();
 boolean result=udao.checkLogin(id,pw);
 String email="sodeo123";

 if(result==true){
     String manager=udao.getEmail(id);
     System.out.printf("manager",manager);
     if(manager.equals(email)==true){
         //관리자로
         System.out.println("로그인 성공!:세션에 값 저장!");
         session.setAttribute("id", id);
         response.sendRedirect("Board_Main.jsp");
     }else{
         //사용자로
         System.out.println("로그인 성공!:세션에 값 저장!");
         session.setAttribute("id", id);
         response.sendRedirect("0.login_main.jsp");
     }
 } else {
     System.out.println("로그인 실패!");
 }
```

관리자 ID를 userDB에 함께 저장함

이메일 값을 반환 받아 비교하여
관리자 인증

○ 잘된 부분

- 게시판 중복 검색 및 페이징 처리 완료.
- 세션을 이용한 로그인 구현.
- CSS를 이용한 프론트 엔드 구현

○ 아쉬운 부분

- 세세한 디테일적 기능 미구현.

- 심미적 아쉬움.

- 개발 계획 수립 미흡.

(전체적인 구조적 설계 미이행)

THANK YOU

