

생활코딩  
**Node.js** 노드제이에스 프로그래밍

**Gachon Shop – gcshop** 프로젝트

2024.2학기 10주

과제 관련

**page**메소드에서 **left join**에서 **topic[0].id** 이야기

**delete**에서 **else** 멈추는 거 **else** 또는 **return**을 안해서

**delete** 다양한 링크에서 안 되는거

**Column** 명 **DB** 스키마

## 6. 게시판 - 테이블

**boardtype**

필드명	데이터형	
type_id	int auto increment	PK
title	varchar(200) not null	
description	varchar(400)	
write_YN	varchar(1) not null	일반 사용자가 글을 쓸 수 있도록 하는 게시판. 불만사항 같은 게시판에 이 옵션을 체크
re_YN	varchar(1) not null	답변을 할 수 있는 게시판. 지금은 사용하지 않음.
numPerPage	int	한 페이지에 나 올 글의 개수

## 6. 게시판 - 테이블

**board**

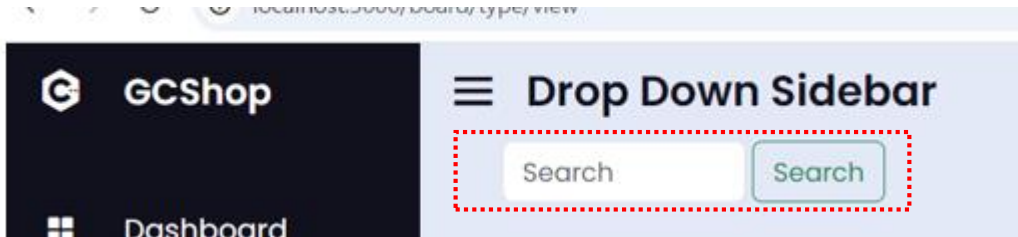
필드명	데이터형	
type_id	int	게시판 타입 id. boardtype 테이블과 연결해 주는 외부 키
board_id	int auto increment	PK
p_id	int	답변 기능을 추가할 때 사용하려고 만든 컬럼. 현재는 사용하지 않음
loginid	varchar(10) NOT NULL	
password	varchar(20)	
title	varchar(20) NOT NULL	
date	varchar(20)	
content	text	

## 6. 게시판 - MVC 모델

boardRouter	router URL	Control(Logic)	Template	DB
/board/	/type/view	board.typeview	./views/boardtype.ejs	boardtype
	/type/create	board.typecreate	./views/boardtypeCU.ejs	boardtype
	/type/create_process	board.typecreate_process		boardtype
	/type/update/:typeid	board.typeupdate	./views/boardtypeCU.ejs	boardtype
	/type/update_process	board.typeupdate_process		boardtype
	/type/delete/:typeid	board.typedelete_process		boardtype
	/view/:typeid/:pNum	board.view	./views/board.ejs	board
	/create/:typeid	board.create	./views/boardCRU.ejs	board
	/create_process	board.create_process		board
	/detail/:boardId/:pNum	board.detail	./views/boardCRU.ejs	board
	/update/:boardId/:typeid/:pNum	board.update	./views/boardCRU.ejs	board
	/update_process	board.update_process		board
	/delete/:boardId/:typeid/:pNum	board.delete_process		board

## 6. 게시판 - 게시판 종류 화면 구성

0) 추후에 사용할 Search input 태그와 버튼 추가



```

<section class="home-section">
  <div class="home-content">
    <i class='bx bx-menu'></i>
    <span class="text">Drop Down Sidebar</span>
  </div>
  <div>
    <form class="d-flex" role="search" action="/shop/search" method="post">
      <input class="form-control me-1" style = "width : 150px;margin-left : 50px;" type="search"
        placeholder="Search" aria-label="Search" name="search">
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
  <div class="home-content2">
    <%- include(body)%>
  </div>
</section>

```

mainframe.ejs의 section 부분에 다음 빨간 박스의 코드를 추가하세요.

## 6. 게시판 - 게시판 종류 화면 구성

1) 게시판 종류 ➔ 관리자(MNG)만 게시판 종류를 입력할 수 있다.

① 게시판 종류 검색 : DB Admin > board type CRUD 를 클릭하면 오른쪽 화면의 게시판 종류 목록이 나옴

← → ↺ localhost:3000/board/type/view ☆ 👤 다시 실행하여 업데이트하세요. ⋮

**GCShop**

- Dashboard
- Category
- Board
- DB Admin
  - Table Create
  - code CRUD
  - product CRUD
  - person CRUD
  - board type CRUD**
- Setting

≡ Drop Down Sidebar

Search Search

### boardtype.ejs

제목	페이지당 글 수	설명	일반 사용자 쓰기	댓글		
Q & A	2	질의 응답 전용 게시판	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
공지사항	2	쇼핑몰 관련 공지사항 게재	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
상품 후기	2	고객들의 상품 후기 관련	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
고객 불만	2	고객 불만에 관한 글	Y	N	<a href="#">수정</a>	<a href="#">삭제</a>
환불 및 취소 관련	2	환불 요청에 관한 게시글	Y	Y	<a href="#">수정</a>	<a href="#">삭제</a>

게시판 종류 추가

## 6. 게시판 - 게시판 종류 화면 구성

1) 게시판 종류 ➔ 관리자(MNG)만 게시판 종류를 입력할 수 있다.

② 자료가 없으면 ㉠ 박스의 내용과 같이 나오게 해야 하고 ㉡ 박스와 같이 Board 메뉴에 아무것도 나타나지 않음

← → ↻ localhost:3000/board/type/view ☆ 👤 다시 실행하여 업데이트하세요. ⋮

**GCShop**

- Dashboard
- Category ▾
- Board** ▴
- DB Admin ▾
- Setting ▾

### Drop Down Sidebar

제목	페이지당 글 수	설명	일반 사용자 쓰기	댓글
<div> <b>㉠</b> 자료가 없으니 게시판 종류 추가 버튼으로 코드를 추가해 주십시오. </div> <div> <input type="button" value="게시판 종류 추가"/> </div>				



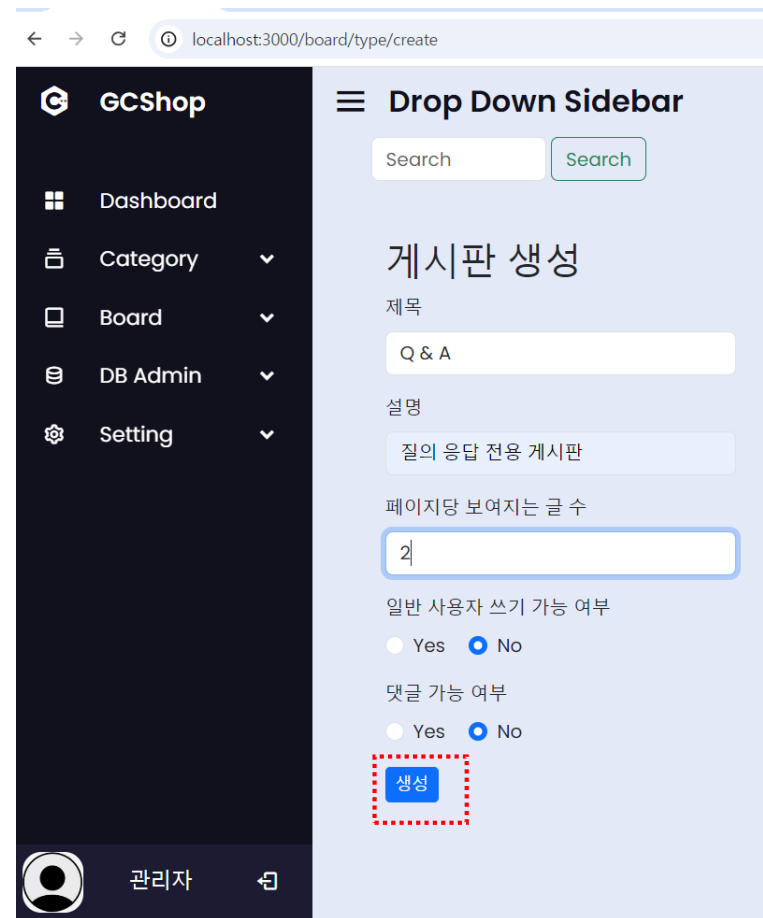
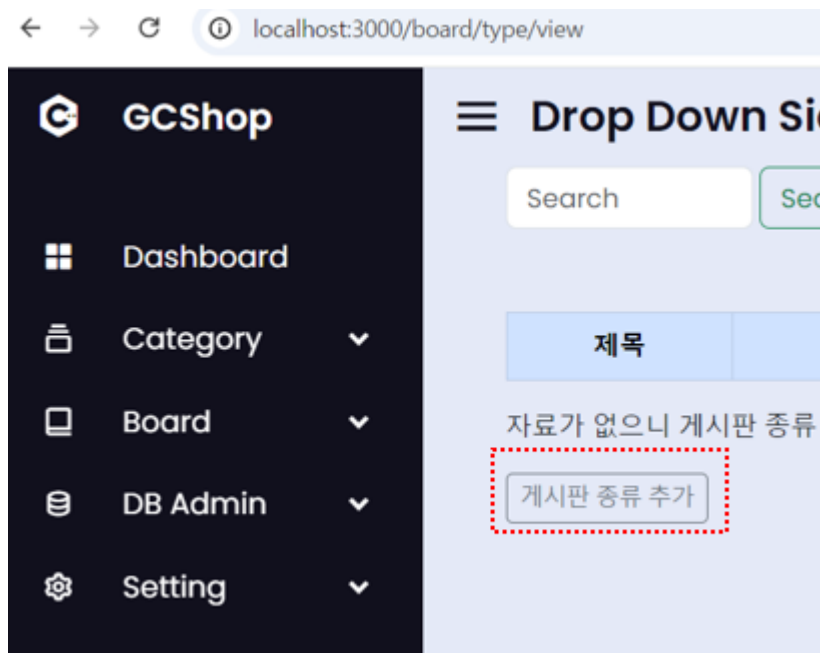
## 6. 게시판 - 게시판 종류 화면 구성

1) 게시판 종류 ➔ 관리자(MNG)만 게시판 종류를 입력할 수 있다.

③ 게시판 종류 추가 버튼을 클릭하면 우측과 같이 게시판 종류를 추가할 수 있는 화면이 나온다.

boardtype 테이블에 자료를 입력한 후에는 /board/type/view로 redirect

boardtypeCU.ejs



## 6. 게시판 - 게시판 종류 화면 구성

1) 게시판 종류 ➔ 관리자(MNG)만 게시판 종류를 입력할 수 있다.

④ 게시판 종류를 추가하면 Board 메뉴에 추가한 게시판의 제목이 뜬다. 수정 링크 클릭하면 수정 화면이 나온다.

← → ↺ ⓘ localhost:3000/board/type/view ☆ 👤 다시 실행하여 업데이트하세요.

**GCShop**

- Dashboard
- Category ▾
- Board ▴
  - Q & A
- DB Admin ▾
- Setting ▾

### Drop Down Sidebar


제목	페이지당 글 수	설명	일반 사용자 쓰기	댓글		
Q & A	2	질의 응답 전용 게시판	N	N	<a href="#">수정</a>	<a href="#">삭제</a>


## 6. 게시판 - 게시판 종류 화면 구성


1) 게시판 종류 ➔ 관리자(MNG)만 게시판 종류를 입력할 수 있다.


- ⑤ 수정 링크 클릭하면 나오는 화면  
수정하고 버튼 클릭하면  
/board/type/view로 redirect


localhost:3000/board/type/update/9 boardtypeCU.ejs


 **GCShop**


 Dashboard

 Category ▾

 Board ▾

 DB Admin ▾

 Setting ▾

 관리자

≡ Drop Down Sidebar

### 게시판 수정

제목

설명

페이지당 보여지는 글 수

일반 사용자 쓰기 가능 여부  
☐ Yes ☒ No

댓글 가능 여부  
☐ Yes ☒ No

11

## 6. 게시판 - 게시판 종류 화면 구성

1) 게시판 종류 → 관리자(MNG)만 게시판 종류를 입력할 수 있다.

⑥ 삭제 클릭하면 삭제 할건지 물어본다.

localhost:3000 내용:

정말로 삭제하시겠습니까?

확인취소

## 6. 게시판 - 게시판 종류 코딩

1) 게시판 종류 → 좌측 Board 메뉴에 하위 메뉴로 게시판 종류가 나오도록 하는 코드

① 게시판 종류 검색

mainFrame.ejs → boardtype 테이블을 읽어서 메뉴를 구성할 수 있게 수정

```
<li>
  <div class="icon-link">
    <a href="#">
      <i class='bx bx-book-alt'></i>
      <span class="link_name">Board</span>
    </a>
    <i class='bx bxs-chevron-down arrow' ></i>
  </div>
  <ul class="sub-menu">
    <li><a class = "link_name" href="#" >Board</a></li>
    <% var k = 0;
      while(k < boardtypes.length) { %>
      <li><a href="/board/view/<%=boardtypes[k].type_id%>/1"><%-boardtypes[k].title%></a></li>
      <% k++;} %>
    </ul>
  </li>
```

mainFrame.ejs → 모든 메소드는 mainFrame.ejs에 boardtypes를 넘겨주어야 함

```
① var sql1 = `select * from boardtype;`  
  var sql2 = `select * from code;`  
  db.query(sql1 + sql2, (error, results) => {  
    var context = {  
      /***** mainFrame.ejs에 필요한 변수 *****/  
      who : name,  
      login : login,  
      body : 'code.ejs',  
      cls : cls,  
      ② boardtypes : results[0],  
      /***** code.ejs에 필요한 변수 *****/  
      codes : results[1]  
    };  
    req.app.render('mainFrame', context, (err, html) => {  
      res.end(html); })  
  });
```

- ① 두 개의 query를 한꺼번에 실행하는 방법 : 두 query를 세미 콜론으로 연결하여 실행. query 끝에 세미콜론 추가
- ② mainFrame.ejs에서 필요한 boardtypes를 context에 실어 보내는 코드. 모든 메소드에 boardtype을 select하여 그 결과를 context에 추가하는 코드가 필요.

두 query 중에서 먼저 실행된 query가 results[0], 두 번째로 실행된 query가 results[1]이다.

mainFrame.ejs → 모든 메소드는 mainFrame.ejs에 boardtypes를 넘겨주어야 함

```
db.query(`select * from boardtype`,(err,results) => {  
  var context = {  
    /***** menuForManager.ejs에 필요한 변수 *****/  
    who : name,  
    login : login,  
    body : 'codeCU.ejs',  
    cls : cls,  
    boardtypes : results,  
    /***** codeCU.ejs에 필요한 변수 *****/  
    cu : 'C'  
  };  
  req.app.render('mainFrame',context, (err, html)=>{  
    res.end(html); })  
})
```

※ query 메소드에 복수 개의 query를 실행시키려면 db.js파일에 코드를 추가해야 함.

```
var mysql = require('mysql');
var db = mysql.createConnection({
  host : 'localhost',
  user : 'nodejs',
  password : 'nodejs',
  database : 'webdb2024',
  multipleStatements: true
});

db.connect();
module.exports = db;
```



## 6. 게시판 - 게시판 종류 코딩

### 1) 게시판 종류

#### ② 게시판 종류 입력

boardtypeCU.ejs : 라디오 버튼 처리

```
<label class="form-label" >일반 사용자 쓰기 가능 여부 </label>
<div class="mb-3">
  <div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="write_YN" id="write_YN1" value="Y">
    <label class="form-check-label" for="write_YN1"> Yes </label>
  </div>
  <div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="write_YN" id="write_YN2" value="N" checked>
    <label class="form-check-label" for="write_YN2"> No </label>
  </div>
</div>
```

## 6. 게시판 - 게시판 종류 코딩

### 1) 게시판 종류

#### ③ 게시판 종류 수정

boardtypeCU.ejs : 라디오 버튼 처리

```
<% var checkedY = '';  
var checkedN = '';  
if(boardtype[0].write_YN === 'Y') {checkedY = 'checked'}  
else {checkedN = 'checked'} %>  
  
<label class="form-label" >일반 사용자 쓰기 가능 여부 </label>  
<div class="mb-3">  
  <div class="form-check form-check-inline">  
    <input class="form-check-input" type="radio" name="write_YN" id="write_YN1" value="Y" <%=checkedY%>>  
    <label class="form-check-label" for="write_YN1"> Yes </label>  
  </div>  
  <div class="form-check form-check-inline">  
    <input class="form-check-input" type="radio" name="write_YN" id="write_YN2" value="N" <%=checkedN%>>  
    <label class="form-check-label" for="write_YN2"> No </label>  
  </div>  
</div>
```

## 6. 게시판 - 게시판에 대한 조건

### 2) 게시판

- ① 로그인 하지 않은 사람과 경영자(CEO)는 게시판을 볼 수만 있다.
- ② 로그인 한 사용자(CST)는 모든 게시물을 볼 수는 있으나 본인이 작성한 게시물만 update, delete 가능하다.
- ③ 로그인 한 사용자(CST)는 게시판 타입이 쓸 수 있는 게시판(boardtype 테이블에서 write\_YN 필드가 Y인 게시판)에 대해서만 게시물을 쓸 수 있다.
- ⑤ 로그인 한 사용자(CST)는 게시물을 쓸 때에는 비번 입력 칸이 있어 게시물 수정할 때 비번을 비교 하여 일치할 때에만 수정 할 수 있다.
- ⑥ 관리자(MNG)는 모든 게시물을 보고 쓰고 갱신하고 삭제할 수 있다.
- ⑦ 모든 게시판에는 paging 기능이 있다.

관련 SQL :  
SELECT \*  
FROM board  
ORDER BY date desc, board\_id  
LIMIT 2 OFFSET 2;

OFFSET : 매 페이지의 시작 위치를 의미, 1page는 OFFSET이 0  
((OFFSET x LIMIT) + 1) 번째 레코드 부터 시작해서 2개를 select

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 하지 않은 사람, 조회만 가능

The screenshot displays the GCShop announcement board interface. On the left is a dark sidebar with the GCShop logo and navigation links: Dashboard, Category, Board, and Setting. The main content area has a 'Drop Down Sidebar' with a search bar and the title '공지사항' (Announcement). Below the title is a table with columns for '작성자' (Author), '제목' (Subject), and '날짜' (Date). The first row shows '관리자' (Admin) as the author, '날씨' (Weather) as the subject, and '2024.11.04 : 20시 58분 6초' (2024.11.04 : 20:58:06) as the date. The '제목' cell is highlighted with a red box. Below the table is a pagination bar showing '1'. To the right of the table is a large arrow pointing to a detailed view of the announcement. This view includes a search bar, a title '제목' (Subject) with the value '날씨' (Weather), an author '작성자' (Author) with the value '관리자' (Admin), a date '작성일' (Date) with the value '2024.11.04 : 20시 58분 6초', and a content area '내용' (Content) with the text '날씨가 추워져요' (The weather is getting colder). A blue '목록' (List) button is at the bottom right.

작성자	제목	날짜
관리자	날씨	2024.11.04 : 20시 58분 6초

1

제목  
날씨

작성자  
관리자

작성일  
2024.11.04 : 20시 58분 6초

내용  
날씨가 추워져요

목록

Guest

## 6. 게시판 - 게시판 화면

## 2) 게시판 - 로그인 한 일반 사용자(CST)

제목	페이지당 글 수	설명	일반 사용자 쓰기	댓글		
Q & A	2	질의 응답 전용 게시판	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
공지사항	2	쇼핑몰 관련 공지사항 개재	N	N	<a href="#">수정</a>	<a href="#">삭제</a>

공지사항은 일반 사용자 글쓰기가 N으로 되어 있어 일반 사용자는 글을 쓸 수 없음

## 6. 게시판 - 게시판 화면

2) 게시판 - 로그인 한 일반 사용자(CST)    공지사항은 일반 사용자 글쓰기가 N으로 되어 있어 일반 사용자는 글을 쓸 수 없음

공지사항

작성자	제목	날짜
관리자	날씨	2024.11.04 : 20시 58분 6초

1

← → ↻

localhost:3000/board/detail/4/1

🔍 ☆ 👤 다시 실행

GCShop

Dashboard

Category ▾

Board ▾

Purchase List

Cart

Setting ▾

Drop Down Sidebar

Search

Search

제목

날씨

작성자

관리자

작성일

2024.11.04 : 20시 58분 6초

내용

날씨가 추워져요

목록

👤 고객1

+

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

제목	페이지당 글 수	설명	일반 사용자 쓰기	댓글		
Q & A	2	질의 응답 전용 게시판	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
공지사항	2	쇼핑몰 관련 공지사항 개재	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
상품 후기	2	고객들의 상품 후기 관련	N	N	<a href="#">수정</a>	<a href="#">삭제</a>
고객 불만	2	고객 불만에 관한 글	Y	N	<a href="#">수정</a>	<a href="#">삭제</a>
환불 및 취소 관련	2	환불 요청에 관한 게시글	Y	Y	<a href="#">수정</a>	<a href="#">삭제</a>

고객 불만과 환불 및 취소 관련 게시판은 일반 사용자 쓰기가 Y으로 되어 있어 일반 사용자도 글을 쓸 수 있음

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

The screenshot shows the GcShop web application interface. The browser address bar displays 'localhost:3000/board/view/12/1'. The page features a dark sidebar on the left with the 'GcShop' logo and navigation links: Dashboard, Category, Board, Purchase List, Cart, and Setting. The main content area has a 'Drop Down Sidebar' with a search bar and a title '고객 불만' (Customer Complaint). Below the title is a table with three columns: '작성자' (Author), '제목' (Title), and '날짜' (Date). A blue '글쓰기' (Write) button is located to the right of the table. At the bottom of the page, a user profile section shows a circular avatar and the text '고객1' (Customer 1). A red dashed box highlights the '글쓰기' button and the user profile section. Below the table, a text block states: '고객 불만과 환불 및 취소 관련 게시판은 일반 사용자 쓰기가 Y으로 되어 있어 일반 사용자도 글을 쓸 수 있음' (Customer complaint and refund/cancellation related boards allow general users to write posts).

← → ↻ ⓘ localhost:3000/board/view/12/1 🔍 ☆ 👤 다시 실행하여 업데이트하세요.

**GcShop**

- Dashboard
- Category
- Board
- Purchase List
- Cart
- Setting

**Drop Down Sidebar**

Search Search

### 고객 불만

작성자	제목	날짜
-----	----	----

글쓰기

고객 불만과 환불 및 취소 관련 게시판은 일반 사용자 쓰기가 Y으로 되어 있어 일반 사용자도 글을 쓸 수 있음

고객1



## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

고객 불만 작성

제목

사이트가 자주 다운되요

작성자

고객1

내용

너무 자주 다운되요

강신삭제시 비밀번호

1111

생성

### 고객 불만

작성자	제목	날짜
고객1	사이트가 자주 다운되요	2024.11.04 : 21시 19분 23초

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

The screenshot shows the GCShop application interface. On the left is a dark sidebar with the GCShop logo and a menu: Dashboard, Category, Board, Purchase List, Cart, and Setting. The main content area has a light blue header with a 'Drop Down Sidebar' and a search bar. Below the header, the title '고객 불만' (Customer Complaint) is displayed. A table lists two complaints:

작성자	제목	날짜
고객2	싸이트가 너무 느려요	2024.11.04 : 21시 21분 11초
고객1	싸이트가 자주 다운되요	2024.11.04 : 21시 19분 23초

Below the table is a pagination bar showing '1'. A '글쓰기' (Write) button is in the top right corner. At the bottom left, a user profile for '고객2' is shown with a red dashed box around it.

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

The screenshot shows the GCSHOP board detail page for a logged-in user. The browser address bar shows the URL `localhost:3000/board/detail/6/1`. The page features a dark sidebar on the left with navigation links: Dashboard, Category, Board, Purchase List, Cart, and Setting. The main content area has a 'Drop Down Sidebar' with a search bar. The board details are displayed in a light blue box with the following fields:

- 제목 (Title): 싸이트가 너무 느려요
- 작성자 (Author): 고객2 (highlighted with a red dashed box)
- 작성일 (Created Date): 2024.11.04 : 21시 21분 11초
- 내용 (Content): 너무 느려요

At the bottom right of the board details, there are three buttons: 목록 (List), 수정 (Edit), and 삭제 (Delete). The 수정 and 삭제 buttons are highlighted with a red dashed box. At the bottom left of the sidebar, the user's profile is shown with the name '고객2' and a logout icon, also highlighted with a red dashed box.

작성자와 login 한 사람이 일치해서 수정 삭제 가능

## 6. 게시판 - 게시판 화면

## 2) 게시판 - 로그인 한 일반 사용자(CST)

The screenshot shows the GCSHOP board detail page at localhost:3000/board/detail/5/1. The page has a dark sidebar on the left with the GCSHOP logo and navigation links: Dashboard, Category, Board, Purchase List, Cart, and Setting. The main content area is titled 'Drop Down Sidebar' and contains a search bar. Below the search bar, the post details are displayed:
 

- 제목 (Title): 싸이트가 자주 다운되요
- 작성자 (Author): 고객1 (highlighted with a red dashed box)
- 작성일 (Write Date): 2024.11.04 : 21시 19분 23초
- 내용 (Content): 너무 자주 다운되요

 A blue '목록' (List) button is located at the bottom right of the content area, also highlighted with a red dashed box. At the bottom left of the sidebar, the user's profile is shown with a circular icon and the name '고객2' (highlighted with a red dashed box).

작성자와 login 한 사람이 일치하지 않아서 수정 삭제 불가

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)

localhost:3000/board/update/6/12/1

GCShop

Dashboard

Category

Board

Purchase List

Cart

Setting

Drop Down Sidebar

Search

Search

고객 불만 수정

제목

사이트가 너무 느려요. 오류도 많아요

작성자

고객2

내용

너무 느려요

갱신삭제시 비밀번호

1111

수정

비밀번호 체크 해야만 수정 가능

고객2

localhost:3000 내용:

비밀번호가 일치하지 않습니다.

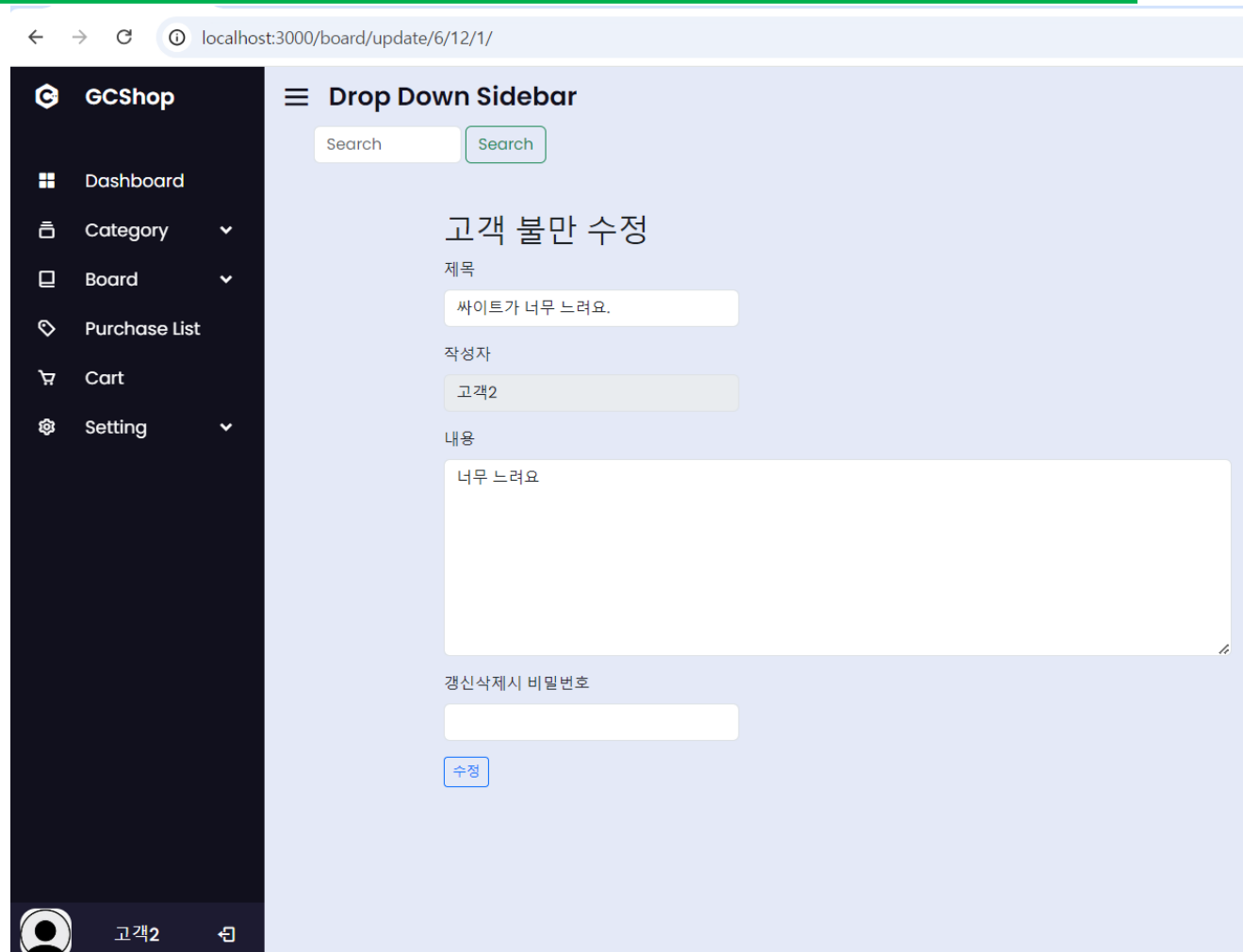
확인

## 6. 게시판 - 게시판 화면

## 2) 게시판 - 로그인 한 일반 사용자(CST)

localhost:3000 내용:  
비밀번호가 일치하지 않습니다.

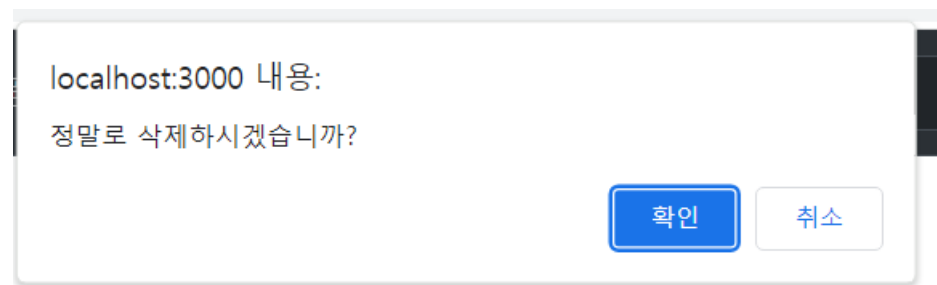
확인



```
res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
res.end(`<script language=JavaScript type="text/javascript">alert("비밀번호가 일치하지 않습니다.")
<!--
  setTimeout("location.href='http://localhost:3000/board/update/${sntzedBoardId}/${sntzedTypeId}/${sntzedpNum}/"', 1000)
  //-->
</script>`)
```

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 로그인 한 일반 사용자(CST)



삭제 버튼 클릭 시 확인 메시지

## 6. 게시판 - 게시판 화면

### 2) 게시판 - 관리자(MNG)

← → ↺ localhost:3000/board/view/12/1 🔍 ☆ 👤 다시 실행하여 업데이트하세요.

GCShop

Dashboard

Category ▾

Board ▾

DB Admin ▾

Setting ▾

Drop Down Sidebar

### 고객 불만

작성자	제목	날짜
고객2	싸이트가 너무 느려요.	2024.11.04 : 23시 5분 41초
고객1	싸이트가 자주 다운되요	2024.11.04 : 21시 19분 23초

1

관리자

모든 게시판에 대해 글쓰기 가능



## 6. 게시판 - 게시판 화면

### 2) 게시판 - 관리자(MNG)

localhost:3000/board/detail/6/1

GCShop

Dashboard

Category

Board

DB Admin

Setting

Drop Down Sidebar

Search

제목

싸이트가 너무 느려요.

작성자

고객2

작성일

2024.11.04 : 23시 5분 41초

내용

너무 느려요

목록 수정 삭제

관리자

모든 글에 대해 수정 삭제 가능. 관리자는 수정할 때 비번이 필요없다.

## 6. 게시판 - 게시판 화면

### 2) 게시판 - paging

#### 고객 불만

작성자	제목	날짜
관리자	인터넷 접속	2024.11.04 : 23시 23분 8초
관리자	불만을 드려 죄송합니다.	2024.11.04 : 23시 22분 31초
고객1	웹사이트가 자꾸 다운되요	2024.11.04 : 23시 20분 39초

[글쓰기](#)[1](#) [2](#)

paging기능

## 6. 게시판 - 페이징 코드

## 2) 게시판 - paging

페이지당 글 수	설정
4	학생
2	직원

numPerPage

board.js

```

view : (req,res)=>{
  var {name, login, cls} = util.authIsOwner(req,res);
  var sntzedTypeId = sanitizeHtml(req.params.typeId);
  var pNum = req.params.pNum;
  var sql1 = `select * from boardtype;` // results[0]
  var sql2 = ` select * from boardtype where type_id = ${sntzedTypeId};` // results[1]
  var sql3 = ` select count(*) as total from board where type_id = ${sntzedTypeId};` // results[2]
  db.query(sql1 + sql2 + sql3, (error,results)=>{
    /*****페이지 기능 구현 *****/
    var numPerPage = results[1][0].numPerPage;
    var offs = (pNum-1)*numPerPage;
    var totalPages = Math.ceil(results[2][0].total / numPerPage);
    db.query(`select b.board_id as board_id, b.title as title, b.date as date, p.name as name
              from board b inner join person p on b.loginid = p.loginid
              where b.type_id = ? and b.p_id = ? ORDER BY date desc, board_id desc LIMIT ? OFFSET ?`,
              [sntzedTypeId, 0, numPerPage, offs], (err,boards)=>{

```

## 6. 게시판 - 페이징 코드

### 2) 게시판 - paging

board.ejs

```
<% var j = 1;
    while (j <= totalPages) {
        if(j == pNum){ %>
            <span style="font-weight : bold;"><%=j%></span> &nbsp;&nbsp;&nbsp;
        <% } else { %>
            <span><a href="/board/view/<%=btname[0].type_id%>/<%=j%>"><%=j%></a></span> &nbsp;&nbsp;&nbsp;
        <% }
        j++
    } %>
```