

# 프로젝트 포트폴리오

---

이름 : 김희원

기술 스택 : Back End / Front End

연락처 : 010-5571-7187

이메일 : khww0301@gmail.com

Git 주소 :

# 목차

table of contents

- 1 도서쇼핑몰
- 2 뮤직플레이어
- 3 게시판만들기
- 4 학사관리시스템



# 1

도서쇼핑몰

---

## 프로젝트 개요

프로젝트 목적

주요 기능



- 온라인에서 책을 구매하고, 장바구니 및 주문 배송 관리 기능을 제공하는 도서 쇼핑몰을 개발
- 사용자에게 편리한 도서 구매 경험을 제공하고, 관리자에게 효율적인 관리 기능을 제공

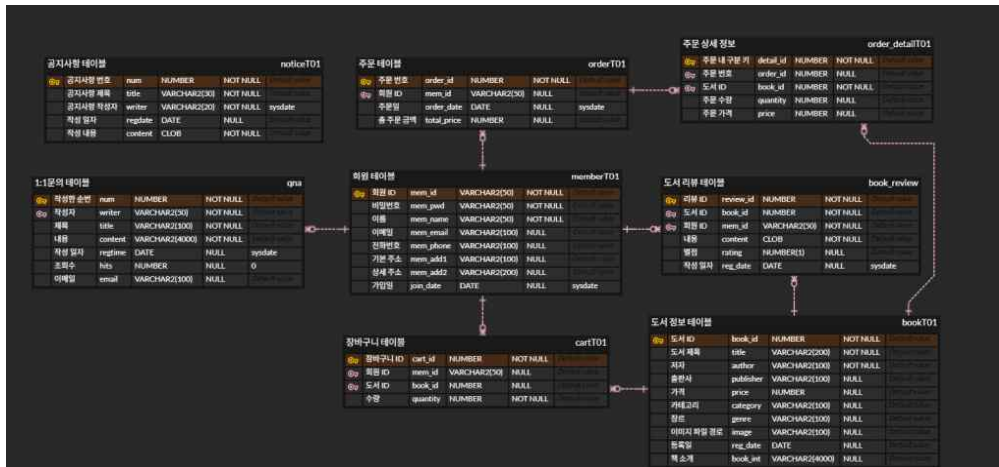
## 프로젝트개요

프로젝트 목적

주요기능

- 회원가입 및 로그인 : 개인 정보를 관리 및 추적
- 장바구니 : 선택한 도서를 장바구니에 담고, 수량 변경 및 삭제
- 주문하기 : 장바구니에 담긴 도서를 주문
- 관리자페이지 : 고객센터, 공지사항, QNA 등 고객 관리 기능

# 프로젝트 - 도서쇼핑몰 (기획 및 설계)



## Log-In



## 로그인

아이디

아이디를 입력하세요.

패스워드

패스워드를 입력하세요.

로그인

구글로 로그인

페이스북으로 로그인

깃허브로 로그인

패스워드를 잊어버리셨나요?

계정이 없습니까? 가입하세요

## 로그인

1. 도서 쇼핑몰 회원 접속 - 로그인

사용자가 입력한 값과 MemberT02 테이블 ID, 패스워드 값을 비교하여 일치한다면 Session에 저장

```

<%
request.setCharacterEncoding("utf-8");

String memValues[] = request.getParameterValues("memValues");
MemberDTO memDTO = new MemberDTO(memValues[0], memValues[1]);
MemberDAO memDAO = MemberDAO.getInstance();
String rst = memDAO.memLog(memDTO);

if (rst.contains("mem_id")) {
    session.setAttribute("mem_id", memValues[0]);
    session.setAttribute("mem_pwd", memValues[1]);
}

%>

$(document).ready(function () {
    // 페이지 로드 시 포커스
    const pageId = $("body").attr("id");
    if (pageId === "update") {
        $("#password").focus();
    } else {
        $("#id").focus();
    }
}

// 엔터 입력 시 자동 포커스
$("#input").on("keydown", function (event) {
    if (event.key === "Enter") {
        event.preventDefault();

        const allInputs = $(".input");
        const currentIndex = allInputs.index(this);

        if (currentIndex >= 0 && currentIndex < allInputs.length - 1) {
            allInputs.eq(currentIndex + 1).focus();
        } else if (currentIndex === allInputs.length - 1) {
            $("#btn01").focus();
        }
    }
});
});

```





## 로그인

Jquery로 페이지 로드 시 ID에 Focus가  
도록 구현 및 엔터 시 포커스 이동하도록  
구현

Input Name으로 Value 값을 가져와  
회원 테이블과 비교하고 일치할 경우  
Session에 저장  
WebFont로 글꼴 변경



1 장바구니 2 사은품선택 3 주문/결제 4 주문완료

				선택삭제	전체삭제
<input checked="" type="checkbox"/>	 어른의 행복은 조용하다 10% 16020 47800 534P	16,020	- 1 +	<a href="#">배송지 등록 필요 &gt;</a> <a href="#">바로드림</a>	X
1시간 이후 수령 가능					
<input type="checkbox"/>	 급류 10% 12600 44000 420P	12,600	- 1 +	<a href="#">배송지 등록 필요 &gt;</a> <a href="#">바로드림</a>	X
1시간 이후 수령 가능					
<input checked="" type="checkbox"/>	 채식주의자 10% 13500 46000 450P	27,000	- 2 +	<a href="#">배송지 등록 필요 &gt;</a> <a href="#">바로드림</a>	X
1시간 이후 수령 가능					
<input type="checkbox"/>	 초역 부처의 말 10% 16020 47800 534P	48,060	- 3 +	<a href="#">배송지 등록 필요 &gt;</a> <a href="#">바로드림</a>	X
1시간 이후 수령 가능					

## 장바구니

1. 장바구니 테이블 리스트 CRUD
2. 도서 구매 처리

Session에 저장된 ID를 가져와 장바구니  
테이블 리스트를 출력  
수량 변경 및 체크 박스 선택 시  
전체 금액 반영  
주문하기 버튼 클릭 시 체크된 항목 주문  
테이블에 저장

상품 금액 47,800원  
배송비 + 0원  
상품 할인 - 4,780원

결제 예정 금액 43,020원

직립 예정 포인트 1,434P

주문하기

```

minusBtn.addEventListener("click", () => changeQty(-1));
plusBtn.addEventListener("click", () => changeQty(1));
});

checkboxes.forEach(checkbox => {
  checkbox.addEventListener("change", updateSummary);
});

$(document).ready(function () {
  // 페이지 로드 시 포커스
  const pageId = $("body").attr("id");
  if (pageId === "update") {
    $("#password").focus();
  } else {
    $("#id").focus();
  }

  // 엔터 입력 시 자동 포커스
  $(".input").on("keydown", function (event) {
    if (event.key === "Enter") {
      event.preventDefault();

      const allInputs = $(".input");
      const currentIndex = allInputs.index(this);

      if (currentIndex >= 0 && currentIndex < allInputs.length - 1) {
        allInputs.eq(currentIndex + 1).focus();
      } else if (currentIndex === allInputs.length - 1) {
        $("#btn01").focus();
      }
    }
  });
});

```

## 장바구니

DOM addEventListener 사용하여 체크  
된 항목 선별하여 금액 반영 및  
테이블 CRUD 구현

Session에 저장된 ID를 가져와 DAO로 해  
당하는 항목 리스트 가져와 DTO에 담은  
뒤 JSP 스크립틀릿 표현식으로 출력

🏠 홈 | 마이 | 장바구니 | 주문/배송목록



**주문/배송 목록**
"최근 1개월 주문내역입니다."
📄 상세조회

주문내역	0	2	0	0	0
주문/배송내역	준비중	배송중	배송완료	취소	교환/반품
 <b>어른의 형복은 조용하다</b> 16020원	1개	배송중	X		
 <b>채식주의자</b> 27000원	2개	배송중	X		

**유의사항**

- 배송조회는 택배사로 상품건당 후 조회 가능합니다.
- eBook은 구매 후 다운로드 시 이용할 수 있습니다.
- eBook 'sany' 이용내역 조회 및 서비스 예제는 마이문 > sany이용내역 > my이용권 페이지에서 가능합니다.
- 원본인 예치금은 나의 물상 > 예치금에서 원본상장 시 신청계좌로 환불해 드립니다.

## 주문/배송

1. 주문 테이블 리스트 출력
2. 주문 테이블 리스트 삭제

Session ID를 가져와 주문 테이블  
리스트 출력  
X 버튼 클릭 시 주문 ID 값으로 해  
당하는 항목 삭제

```
OrderDAO dao = OrderDAO.getInstance();  
List<OrderDTO> orderList = dao.orderSelOrderDTO(vId);  
MemberDAO memDAO = MemberDAO.getInstance();  
String name = memDAO.memName(memDTO);
```

```
public ArrayList<BookDTO> orderSel(String id) {  
    ArrayList<BookDTO> dtoList = new ArrayList<BookDTO>();  
    try {  
        conn = DBConnection.getConnection();  
        sql = "SELECT b.book_id, b.title, b.price, b.image, od.quantity\r\n"  
            + "FROM memberT02 m\r\n"  
            + "JOIN orderT01 o ON m.mem_id = o.mem_id\r\n"  
            + "JOIN order_detailT01 od ON o.order_id = od.order_id\r\n"  
            + "JOIN bookT01 b ON od.book_id = b.book_id\r\n"  
            + "where m.MEM_ID = ?";  
        pstmt = conn.prepareStatement(sql);  
        pstmt.setString(1, id); // 로그인 id값 넣기  
        rs = pstmt.executeQuery();  
        while(rs.next()) {  
            book_id = rs.getInt("book_id");  
            title = rs.getString("title");  
            price = rs.getInt("price");  
            image = rs.getString("image");  
            quantity = rs.getInt("quantity");  
            dtoList.add(new BookDTO(book_id, title, price, image, quantity));  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        DBConnection.close(rs, pstmt, conn);  
    }  
    return dtoList;  
}
```

## 주문/배송

Session ID를 가져와 주문 테이블  
리스트 출력

X 버튼 클릭 시 주문 ID 값으로  
해당하는 항목 삭제

```
<form method="post" action="deleteOrder.jsp">
  <input type="hidden" name="order_id" value="<%=order.getOrder_id()%>">
  <input type="hidden" name="id" value="<%=id%>">
  <input type="hidden" name="pwd" value="<%=pwd%>">
  <button type="submit" class="delete-btn"
    onclick="return confirm('이 주문을 삭제하시겠습니까?');">X</button>
</form>
```

```
String orderIdStr = request.getParameter("order_id");
int order_id = Integer.parseInt(orderIdStr);

OrderDAO dao = OrderDAO.getInstance();
boolean result = dao.deleteOrder(order_id);

if (result) {
    response.sendRedirect("orderDelivery.jsp?id=" + vId + "&pwd=" + pwd);
} else {
    out.println("<script>alert('삭제 실패'); history.back();</script>");
}
```

## 주문/배송

Session ID를 가져와 주문 테이블  
리스트 출력

X 버튼 클릭 시 주문 ID 값으로  
해당하는 항목 삭제

## 공지 사항

검색

## 공지사항

[공지사항 쓰기]

번호	제목	작성자	등록일
11	공지사항입니다.	고객센터	2025-05-18
10	공지사항	고객센터	2025-05-11
9	공지사항	고객센터	2025-05-11
8	공지사항	고객센터	2025-05-11
7	공지사항	고객센터	2025-05-11
6	공지사항	고객센터	2025-05-11
5	공지사항	고객센터	2025-05-11
4	공지사항	고객센터	2025-05-11
3	공지사항	고객센터	2025-05-11
2	공지사항	고객센터	2025-05-11

## 공지사항

## 1. 공지사항 CRUD 및 페이징 기능

공지사항 리스트 배열에 담아 유기적으로 페이징 기능 실현되도록 구현

```

<%
    ArrayList<NoticeDTO> notices =
        (ArrayList<NoticeDTO>) session.getAttribute("notices");
    session.removeAttribute("notices");
%>
<script>
const notices = [
<%
    for (int i = 0; i < notices.size(); i++) {
        NoticeDTO n = notices.get(i);
%>
        {
            num: <%=n.getNum()%>,
            title: "<%=n.getTitle()%>",
            writer: "<%=n.getWriter()%>",
            date: "<%=n.getRegdate()%>"
        }<%= (i < notices.size() - 1) ? ", " : "" %>
%>
    ]
%>
];

let recordName = {
    totalRecord: notices.length,
    recPerPage: 10,
    pagePerBlock: 10,
    totalPage: 0,
    totalBlock: 0,
    nowPage: 1,
    nowBlock: 1,
    recOfBeginPage: 0,
    pageOfBeginBlock: 0
};

```

## 공지사항

Servlet MVC방식으로 데이터 가져와서  
ArrayList에 값을 담고, 배열에 키와 값으  
로 값을 담아서 페이징 기능 구현

페이징 전체 길이를 ArrayList 길이로  
정하고, 10개의 페이지와 블록으로 구  
성하여 번호 선택 시 유기적으로 리스  
트 변경되도록 구현

```
function createPagination() {
    pagination.innerHTML = '';

    const crtAFirst = document.createElement('a');
    crtAFirst.href = '#';
    crtAFirst.innerHTML = '&laquo;';
    crtAFirst.addEventListener('click', movePrevBlock);
    pagination.appendChild(crtAFirst);

    let startPage = (recordName.nowBlock - 1) * recordName.pagePerBlock + 1;
    let endPage = Math.min(startPage + recordName.pagePerBlock - 1, recordName.totalPage);

    for (let page = startPage; page <= endPage; page++) {
        const crtA = document.createElement('a');
        crtA.href = '#';
        crtA.innerHTML = page;
        crtA.addEventListener('click', function (e) {
            e.preventDefault();
            setActivePage(page);
        });
        pagination.appendChild(crtA);
    }
}
```

## 공지사항

Servlet MVC방식으로 데이터 가져와서  
ArrayList에 값을 담고, 배열에 키와 값으  
로 값을 담아서 페이징 기능 구현

페이징 전체 길이를 ArrayList 길이로  
정하고, 10개의 페이지와 블록으로 구  
성하여 번호 선택 시 유기적으로 리스  
트 변경되도록 구현



## 공지사항

\*\*\*Notice View\*\*\*

[수정] [삭제] [목록보기]

글번호	11	Writer	고적센터
Title	공지사항입니다.	Date	2025-05-18 16:23:17
Contents	<div>오늘 공지사항입니다. &lt;br&gt;좋은 쇼핑하세요.</div>		

## 공지사항

## 1. 공지사항 세부내용 조회

공지사항 클릭 시 해당하는 번호로 테이블 리스트를 출력  
수정 및 삭제 클릭 시 수정페이지 이동  
및 해당하는 항목 삭제

```

<%
while(rs.next()){
    vNo = rs.getInt("num");
    vWriter = rs.getString("writer");
    vTitle = rs.getString("title");
    vRegTime = rs.getString("regdate");
    vContent = rs.getString("content");
}
vTitle = vTitle.replace(" ", "&nbsp;");
vContent = vContent.replace(" ", "&nbsp;").replace("\n", "<br>");
%>

Connection conn = DBConnection.getConnection();
Statement stmt = conn.createStatement();
String temp = request.getParameter("num");
if(temp != null){
    vNo = Integer.parseInt(temp);
}
String sql = "select * from noticeTO1 where num = " + vNo;
ResultSet rs = stmt.executeQuery(sql);
%>
<nav>
<span><a id="list" href="<%=contextPath %>
/boardBean.do?category=noticeUpdate&num=<%=vNo %>">[ 수 정 ]</a></span>
<span><a id="list" href="<%=contextPath %>
/boardBean.do?category=noticeDel&num=<%=vNo %>">[ 삭 제 ]</a></span>
<span><a id="list" href="<%=contextPath %>
/boardBean.do?category=notice">[ 목록보기 ]</a></span>
</nav>

```

```

<tr>
<th class="table_th">글번호</th>
<td class="head_txt"><%=vNo %></td>
<th class="table_th">Writer</th>
<td class="head_txt"><%=vWriter %></td>
</tr>
<tr>
<th class="table_th">Title</th>
<td class="head_txt"><%=vTitle %></td>
<th class="table_th">Date</th>
<td class="head_txt"><%=vRegTime %></td>
</tr>

```

## 공지사항

공지사항 클릭 시 해당하는 번호로  
테이블 리스트를 출력

수정 및 삭제 클릭 시 번호를 넘겨서  
request.parameter 해당하는 항목 삭  
제 및 수정 페이지로 이동

# 2

## 음악플레이어

---

## 프로젝트 개요

프로젝트 목적

주요 기능



- 사용자에게 직관적이고 편리한 음악 재생 기능을 제공
- 웹 인터페이스 설계와 동적 기능 구현 능력을 향상시키기 위한 프로젝트

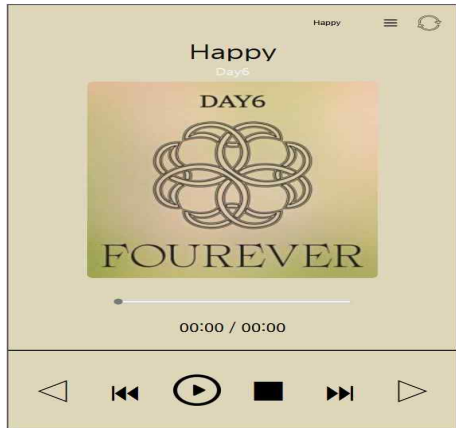
## 프로젝트개요

프로젝트 목적

주요 기능



- 개별 음악 선택 기능
- 다음, 이전, 처음, 마지막 음악 선택 기능
- 음악 재생 및 일시정지, 초기화 기능



## 메인페이지

1. 음악 변동 시 유기적인 변경 구현

음악 변동 시 해당하는 곡에 index를  
반영하여 배열에 값을 불러와  
곡 제목, 가수명, 이미지 등 변경

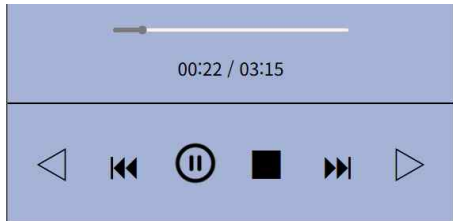


Option 선택 시 해당하는 Value를  
가져와 배열의 값과 비교한 뒤 해당  
하는 index 값을 저장

### 상단부 - select, 재생방법 선택

1. Select부분 Option 선택 시 음악 변경
2. 전체 재생과 한 곡 재생 버튼 구현

재생되는 시간과 끝나는 시간을 비교  
하여 값이 같다면 반복 재생은  
다음곡 으로 넘어가고,  
한 곡 재생은 Loop를 사용하여  
한 곡 재생하도록 구현



RangeBar는 Input type를 range로 하여 구현했으며, 직접 Range를 움직일 수 있도록 Range에 Value를 currntTime에 담아 유기적으로 연동

### 하단부 - 컨트롤러, RangeBar

1. 컨트롤러 구현
2. RangeBar 구현

컨트롤러로 왼쪽부터 처음, 이전으로 이동, 재생&일시정지, 초기화, 다음, 마지막으로 이동 갈 수 있도록 구현



```
function updateImage() {  
  objImg.src = imgArr[currentIndex];  
  objText.value = imgName[currentIndex];  
  objText01.value = singer[currentIndex];  
  document.getElementById("music").value = imgName[currentIndex];  
  
  bgElement.style.backgroundColor = `rgb(${bgColor[currentIndex]})`;  
  objText.style.backgroundColor = `rgb(${bgColor[currentIndex]})`;  
  objText01.style.backgroundColor = `rgb(${bgColor[currentIndex]})`;  
  for (let i = 0; i < btnColor.length; i++) {  
    btnColor[i].style.backgroundColor = `rgb(${bgColor[currentIndex]})`;  
  }  
}
```

1. 앨범 이미지 교체
2. 곡 제목 및 가수명 교체
3. 배경색상 및 버튼 색상 동기화

음악 선택 시 선택된 음악과 배열에 담긴 음악을 비교하여 값이 같다면 해당하는 index를 currntIndex에 담아서 모든 항목을 변경된 음악에 index에 맞게 변경하는 코드

```
myAudio.ontimeupdate = function(){
  if(!isSeeking){
    rangeBar.max = Math.floor(myAudio.duration);
    rangeBar.value = Math.floor(myAudio.currentTime);
  }
  var due = Math.floor(myAudio.duration);
  var ctime = Math.floor(myAudio.currentTime);
  let min1 = Math.floor(ctime / 60);
  let sec1 = Math.floor(ctime % 60);
  let min2 = Math.floor(due / 60);
  let sec2 = Math.floor(due % 60);

  min1 = min1 < 10 ? "0" + min1 : min1;
  min2 = min2 < 10 ? "0" + min2 : min2;
  sec1 = sec1 < 10 ? "0" + sec1 : sec1;
  sec2 = sec2 < 10 ? "0" + sec2 : sec2;

  document.getElementById("info").innerHTML = `${min1}:${sec1} / ${min2}:${sec2}`;
  if((min1 / min2) == 1 && (sec1 / sec2) == 1){
    let image = document.getElementById("sound");
    if(image.src.split('/').pop() == "once.png"){
      myAudio.loop = true;
    }else{
      myAudio.loop = false;
      nextImage();
    }
  }
}
```

1. RangeBar 동기화

2. 현재 시간과 전체 시간을 분 : 초로 구현

사용자가 드래그 사용 시 isSeeking이 true가 되며, 오디오 전체 길이를 rangeBar의 최대값으로 설정한 뒤 현재 재생 위치를 currentTime에 반영

총 재생 시간과 현재 시간을 분 : 초 포맷으로 변환

# 3

## 게시판만들기

---

## 프로젝트개요

프로젝트 목적

주요기능



- 사용자에게는 게시글 등록과 조회를 통한 정보 공유의 편의를 제공
- 게시글을 효율적으로 관리할 수 있도록 기능을 제공

## 프로젝트개요

프로젝트 목적

주요 기능

- 게시글 목록 조회
- 게시글 상세 조회
- 게시글 작성
- 게시글 수정
- 게시글 삭제

## \*\*게시판 목록\*\*

[게시판 내용List] [게시판 목록List] [게시판 쓰기]

번호	제목	글쓴이	등록일	조회	IP
254	최진실인강하세요	최진실	2025-05-13 00:00:00	2	127.0.0.1
253	채시라인강하세요	채시라	2025-05-12 00:00:00	1	127.0.0.1
252	무궁화인강하세요	무궁화	2025-05-11 00:00:00	1	127.0.0.1
251	류명현인강하세요	류명현	2025-05-10 00:00:00	0	127.0.0.1
250	류철나인강하세요	류철나	2025-05-09 00:00:00	1	127.0.0.1
249	공부재인강하세요	공부재	2025-05-08 00:00:00	0	127.0.0.1
248	무궁화인강하세요	무궁화	2025-05-07 00:00:00	1	127.0.0.1
247	채송화인강하세요	채송화	2025-05-06 00:00:00	1	127.0.0.1
246	늘기만인강하세요	늘기만	2025-05-05 00:00:00	1	127.0.0.1
245	공부만인강하세요	공부만	2025-05-04 00:00:00	0	127.0.0.1

1 2 3 4 5 6 7 8 9 10 »

## 게시판 목록

1. BoardV02 테이블에 담긴 내용 출력
2. 페이징 기능 구현

페이징 기능을 사용하여 한 페이지에 10개의 레코드와 10개의 블록을 보여주도록 구현

번호, 제목, 글쓴이, 등록일, 조회 수, IP가 있으며 자세한 사항은 내용을 클릭 시 확인할 수 있도록 구현

## ★★게시판 내용★★

[게시판 내용List] [게시판 목록List] [게시판 쓰기]

[ 수정 ] [ 삭제 ]

Writer	최진실	Date	2025-05-13 00:00:00
Title	최진실안녕하세요		
Contents	네최진실입니다.		

[ 수정 ] [ 삭제 ]

Writer	재시라	Date	2025-05-12 00:00:00
Title	재시라안녕하세요		
Contents	네재시라입니다.		

1 2 3 4 5 6 7 8 9 10 &gt;&gt;

## 게시판 내용 목록

1. 게시판 내용 목록 출력
2. 페이징 기능 구현

게시판 내용은 해당 항목을 선택하면 보여지는 내용을 2개씩 보여주도록 구현  
 페이징 기능을 사용하여 한 페이지에 2개의 내용과 10개의 블록이 보이도록 구현

**\*\*게시판 글쓰기\*\***[\[게시판 내용List\]](#) [\[게시판 목록List\]](#) [\[게시판 쓰기\]](#)

Writer	Orange	Password	*****
Title	Nice		
Email	Orange@naver.com		
Content	제목 하우		

Write

Reset

**게시판 글쓰기****1. 게시판 작성 기능 구현**

게시판을 작성할 때는 작성자, 비밀번호, 제목, 이메일, 내용을 작성  
사용자가 작성한 항목에 내용을 받아 DB와 연동하여 값을 저장



**\*\*비밀번호 확인\*\***[\[게시판 내용List\]](#) [\[게시판 목록List\]](#) [\[게시판 쓰기\]](#)

** 글쓴 날기일 때 입력하신 비밀번호를 입력하여 주십시오 **	
●●●●	
확인	암호복쓰기

비밀 번호 일치 : 클릭시 이동

**비밀번호 확인****1. 수정, 삭제 시 비밀번호 확인 기능 구현**

게시판을 수정 및 삭제할 때 번호와 비밀번호 받아서 사용자가 작성한 비밀번호와 일치한다면 '비밀 번호 일치 : 클릭 시 이동' 버튼이 나오도록 구현

## \*\*게시판 글수정\*\*

[게시판 내용List] [게시판 목록List] [게시판 쓰기]

글번호	254	조회수	3
Writer	최진실	Date	2025-05-13 00:00:00
Title	최진실안녕하세요		
Email	최진실@rest.com		
Contents	<div> <div>내용이 없습니다.</div> </div>		

Update

Reset

## 게시판 글수정

1. 선택한 번호에 해당하는 DB 출력
2. 수정된 값 DB에 반영

비밀번호 일치 시 글수정 페이지로 이동하여 선택한 번호에 해당하는 내용 출력  
글번호, 조회수, date를 제외한 값 수정한 뒤 버튼 클릭 시 DB에 저장

**\*\*게시판 Chk\*\***

[게시판 내용List] [게시판 목록List] [게시판 쓰기]

Message

회원님님을 수정 완료!

**게시판 Chk페이지****1. 게시판 기능 성공 여부 확인 구현**

게시판 작성, 수정, 삭제가 성공 혹은 실패할 경우 게시판 Chk페이지로 이동하여 성공 여부를 확인

```
<jsp:useBean id="bean" class="boardV02.BoardBean" />
<jsp:setProperty name="bean" property="nowPage" param="nowPage" />
<jsp:setProperty name="bean" property="nowBlock" param="nowBlock" />
<%
    bean.execute();
    int nowPage = bean.getNowPage();
    int nowBlock = bean.getNowBlock();
    int totalPage = bean.getTotalPage();
    int totalBlock = bean.getTotalBlock();
    int pageOfBeginBlock = bean.getPageOfBeginBlock();
    int pagePerBlock = bean.getPagePerBlock();
%>
```

```
<%
    ArrayList<BoardV02DTO> list = bean.getBoardList();
    for (int i = 0; i < list.size(); i++) {
        BoardV02DTO row = list.get(i);
    %>
        <tr id="tr" onclick="valSend(<%=row.getBod_no()%>)">
            <td><%=row.getBod_no()%></td>
            <td><%=row.getBod_subject()%></td>
            <td><%=row.getBod_writer()%></td>
            <td><%=row.getBod_logtime()%></td>
            <td><%=row.getBod_readCnt()%></td>
            <td><%=row.getBod_connip()%></td>
        </tr>
    %>
    }
%>
```

자바빈을 선언하고 setNowPage와 setNowBlock 메서드를 호출

bean.execute 메서드를 통해 게시판 데이터와 페이지 정보를 계산하고 조회

```

<%
ArrayList<BoardVO2DTO> list = bean.getBoardList();
for (int i = 0; i < list.size(); i++) {
    BoardVO2DTO row = list.get(i);
%>
<tr id="tr" onclick="valSend(<%=row.getBod_no()%>)">
    <td><%=row.getBod_no()%></td>
    <td><%=row.getBod_subject()%></td>
    <td><%=row.getBod_writer()%></td>
    <td><%=row.getBod_logtime()%></td>
    <td><%=row.getBod_readCnt()%></td>
    <td><%=row.getBod_connip()%></td>
</tr>
<%
}
%>

```

```

<%
int recPerPage = 2;
int pagePerBlock = 10;

int totalRecord = bodDAO.getTotalRecord();
int totalPage = (int)Math.ceil((double)totalRecord / recPerPage);
int totalBlock = (int)Math.ceil((double)totalPage / pagePerBlock);

int nowPage = 0;
int nowBlock = 0;

if (request.getParameter("nowPage") != null) {
    nowPage = Integer.parseInt(request.getParameter("nowPage"));
}
if (request.getParameter("nowBlock") != null) {
    nowBlock = Integer.parseInt(request.getParameter("nowBlock"));
}

int recOfBeginPage = nowPage * recPerPage;
int pageOfBeginBlock = nowBlock * pagePerBlock;
int begin = recOfBeginPage + 1;
int end = begin + recPerPage - 1;

ArrayList<BoardVO2DTO> list = bodDAO.getPagedList(begin, end);
for (BoardVO2DTO dto : list) {
%>

```

페이지당 게시글 수와 블럭당 페이지 수를 기준으로 현재 페이지 및 블럭의 범위를 계산

begin, end 값은 현재 페이지에서 보여줄 게시글 번호 범위로 DAO의 getPagedList(begin, end) 메서드 호출에 사용