

# JSP 게시판 강좌 따라하기 10강

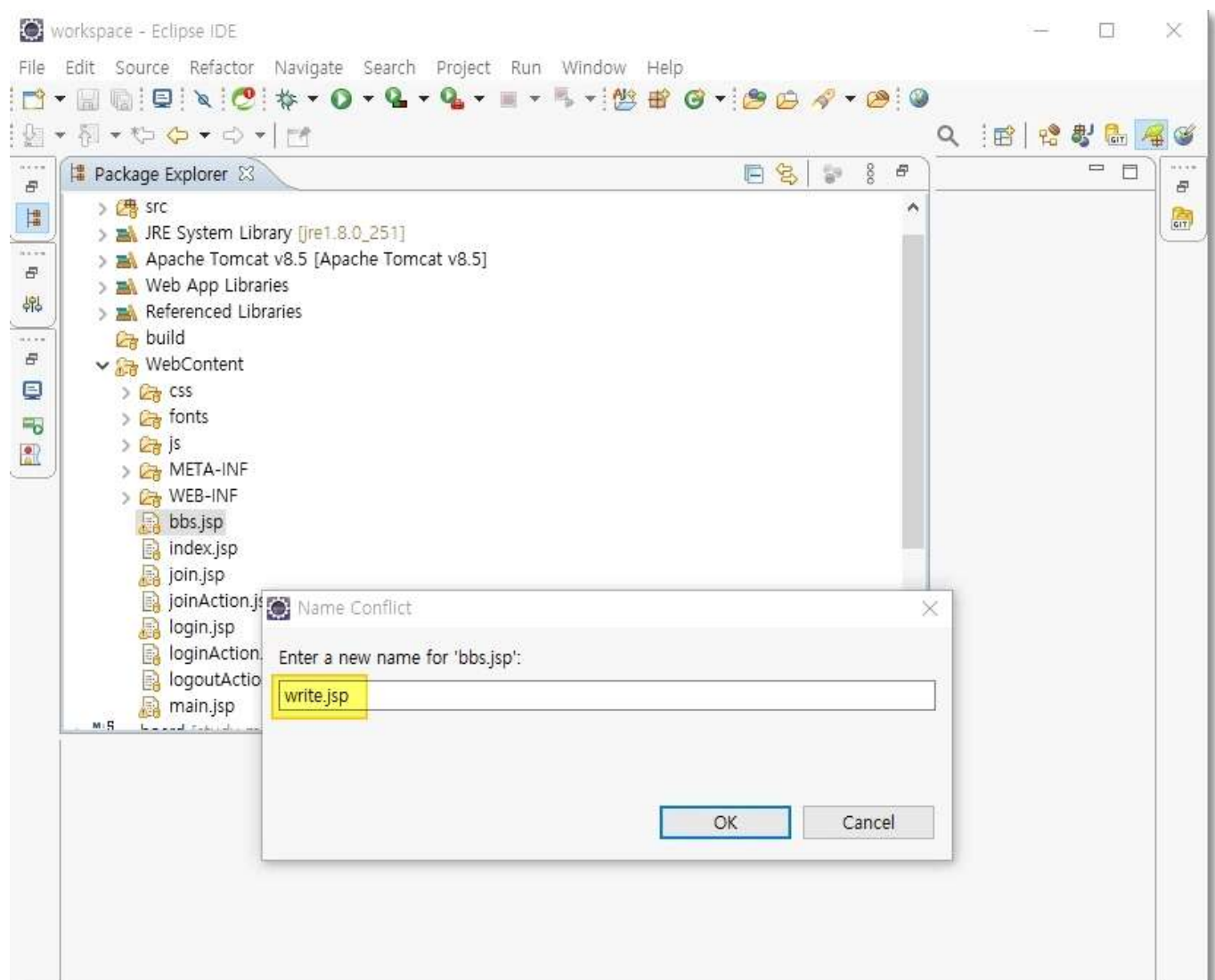
## 게시판 글쓰기 기능 구현하기

---

게시판 페이지에서 로그인을 한 사람들은 '글쓰기' 기능을 사용할 수 있도록 실제 사용자가 입력한 글 제목과 글 내용을 전달 받아 데이터베이스에 실제로 저장되는 기능을 구현해보겠습니다

---

# write.jsp



기존에 있던 'bbs.jsp' 를 복사하여 'write.jsp' 페이지를 생성합니다

```
81 <!-- 게시판 글쓰기 양식 영역 시작 -->
82 <div class="container">
83   <div class="row">
84     <form method="post" action="writeAction.jsp">
85       <table class="table table-striped" style="text-align: center; border: 1px solid #dddddd">
86         <thead>
87           <tr>
88             <th colspan="2" style="background-color: #eeeeee; text-align: center;">게시판 글쓰기 양식</th>
89           </tr>
90         </thead>
91         <tbody>
92           <tr>
93             <td><input type="text" class="form-control" placeholder="글 제목" name="bbsTitle" maxlength="50"></td>
94           </tr>
95           <tr>
96             <td><textarea class="form-control" placeholder="글 내용" name="bbsContent" maxlength="2048" style="height: 350px;"></td>
97           </tr>
98         </tbody>
99       </table>
100       <!-- 글쓰기 버튼 생성 -->
101       <input type="submit" class="btn btn-primary pull-right" value="글쓰기">
102     </form>
103   </div>
104 </div>
105 <!-- 게시판 글쓰기 양식 영역 끝 -->
106
```

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html >
<html >
<head >
<meta charset ="UTF-8">
<!-- 화면 최적화 -->
<meta name ="viewport"content ="width=device-width", initial-scale ="1">
<!-- 루트 폴더에 부트스트랩을 참조하는 링크 -->
<link rel ="stylesheet"href ="css/bootstrap.css">
<title >JSP 게시판 웹 사이트 </title >
</head >
<body >
<%
    // 메인 페이지로 이동했을 때 세션에 값이 담겨있는지 체크
    String userID = null;
    if(session.getAttribute("userID") != null){
        userID = (String)session.getAttribute("userID");
    }
%>
<nav class ="navbar navbar-default"> <!-- 네비게이션 -->
    <div class ="navbar-header">    <!-- 네비게이션 상단 부분 -->
        <!-- 네비게이션 상단 박스 영역 -->
        <button type ="button"class ="navbar-toggle collapsed"
```

```

        data-toggle = "collapse" data-target
        = "#bs-example-navbar-collapse-1"
        aria-expanded = "false">
        <!-- 이 삼줄 버튼은 화면이 좁아지면 우측에 나타난다
-->

        <span class = "icon-bar"></span >
        <span class = "icon-bar"></span >
        <span class = "icon-bar"></span >
    </button >
    <!-- 상단 바에 제목이 나타나고 클릭하면 main 페이지로 이동한다
-->

    <a class = "navbar-brand" href = "main.jsp">JSP 게시판 웹 사이트
</a >

</div >
<!-- 게시판 제목 이름 옆에 나타나는 메뉴 영역 -->
<div class = "collapse navbar-collapse" id
= "bs-example-navbar-collapse-1">
    <ul class = "nav navbar-nav">
        <li ><a href = "main.jsp">메인 </a ></li >
        <li class = "active"><a href = "bbs.jsp">게시판 </a ></li
>

    </ul >
    <%
        // 로그인 하지 않았을 때 보여지는 화면
        if(userID == null){
    %>
    <!-- 헤더 우측에 나타나는 드롭다운 영역 -->
    <ul class = "nav navbar-nav navbar-right">
        <li class = "dropdown">
            <a href = "#" class = "dropdown-toggle"
                data-toggle = "dropdown" role
= "button" aria-haspopup = "true"
                aria-expanded = "false">접속하기 <span class
= "caret"></span ></a >

            <!-- 드롭다운 아이템 영역 -->
            <ul class = "dropdown-menu">
                <li ><a href = "login.jsp">로그인 </a ></li
>

```

```

        <li ><a href ="join.jsp">회원가입 </a ></li
>

        </ul >
    </li >
</ul >
<%
    // 로그인이 되어 있는 상태에서 보여주는 화면
    }else{
%>
<!-- 헤더 우측에 나타나는 드롭다운 영역 -->
<ul class ="nav navbar-nav navbar-right">
    <li class ="dropdown">
        <a href ="#"class ="dropdown-toggle"
            data-toggle ="dropdown"role
="button"aria-haspopup ="true"
            aria-expanded ="false">회원관리 <span class
="caret"></span ></a >
        <!-- 드롭다운 아이탬 영역 -->
        <ul class ="dropdown-menu">
            <li ><a href ="logoutAction.jsp">로그아웃
</a ></li >
        </ul >
    </li >
</ul >
</div >
</nav >
<!-- 네비게이션 영역 끝 -->

<!-- 게시판 글쓰기 양식 영역 시작 -->
<div class ="container">
    <div class ="row">
        <form method ="post"action ="writeAction.jsp">
            <table class ="table table-striped"style ="text-align:
center; border: 1px solid #dddddd">
                <thead >

```

```

        <tr >
            <th colspan = "2" style
="background-color: #eeeeee; text-align: center;">게시판 글쓰기 양식 </th >
        </tr >
    </thead >
    <tbody >
        <tr >
            <td ><input type = "text" class
="form-control" placeholder = "글 제목" name = "bbsTitle" maxlength = "50"></td >
        </tr >
        <tr >
            <td ><textarea class
="form-control" placeholder = "글 내용" name = "bbsContent" maxlength = "2048" style
="height: 350px;"></textarea ></td >
        </tr >
    </tbody >
</table >
<!-- 글쓰기 버튼 생성 -->
<input type = "submit" class = "btn btn-primary
pull-right" value = "글쓰기">
    </form >
</div >
</div >
<!-- 게시판 글쓰기 양식 영역 끝 -->

<!-- 부트스트랩 참조 영역 -->
<script src = "https://code.jquery.com/jquery-3.1.1.min.js"></script >
<script src = "js/bootstrap.js"></script >
</body >
</html >

```

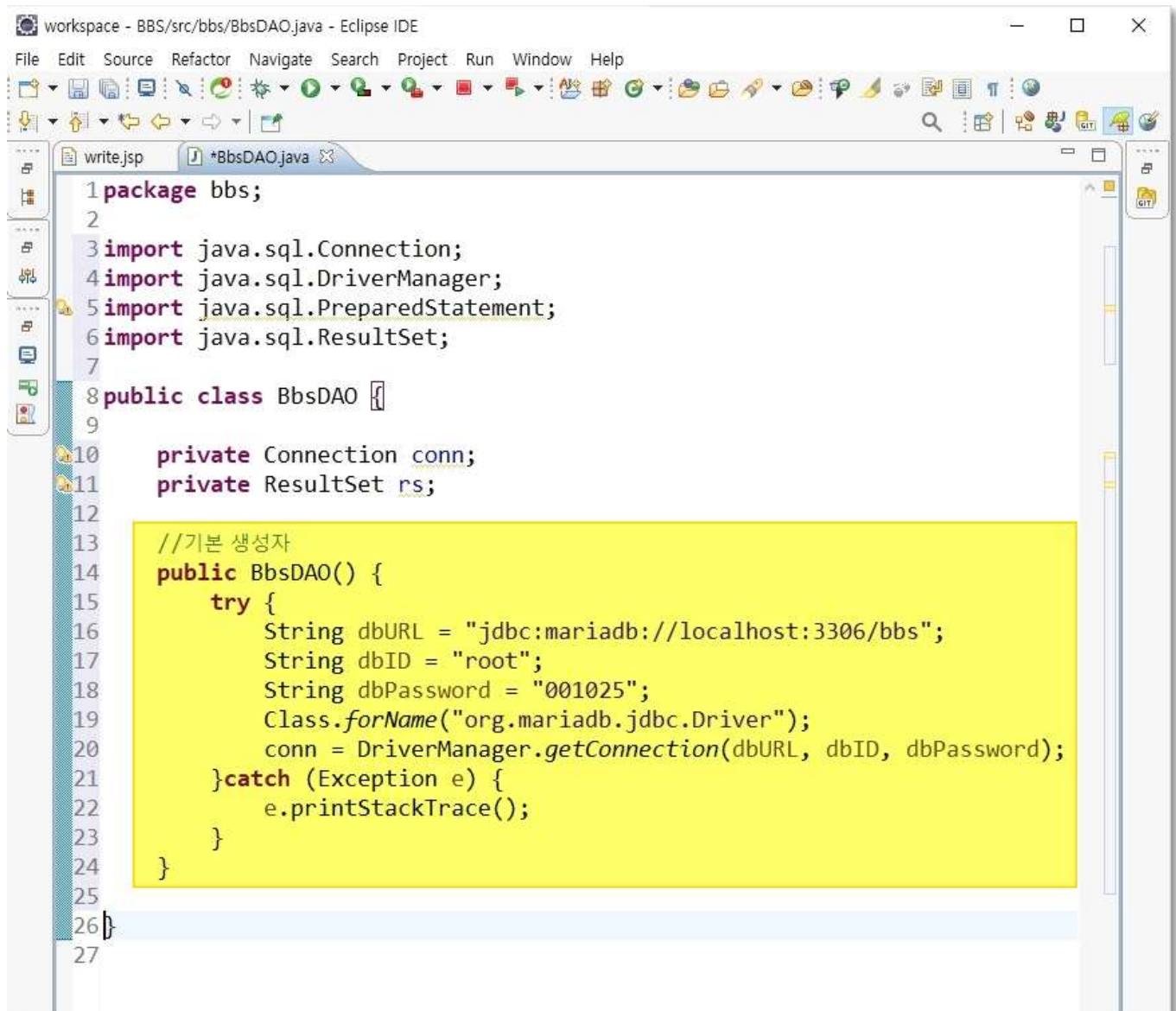
게시판 글쓰기 양식 코드를 추가합니다

'bbs.jsp' 페이지에서 '글쓰기' 버튼을 누르면 글쓰기 양식이 화면에 보여지는 코드입니다

---

# BbsDAO

'bbs' 패키지에서 'BbsDAO' 클래스를 생성합니다



```
1 package bbs;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7
8 public class BbsDAO {
9
10     private Connection conn;
11     private ResultSet rs;
12
13     //기본 생성자
14     public BbsDAO() {
15         try {
16             String dbURL = "jdbc:mariadb://localhost:3306/bbs";
17             String dbID = "root";
18             String dbPassword = "001025";
19             Class.forName("org.mariadb.jdbc.Driver");
20             conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
21         } catch (Exception e) {
22             e.printStackTrace();
23         }
24     }
25
26
27 }
```

'UserDAO' 와 비슷하게 기본 생성자 메소드를 가장 먼저 입력합니다

'BbsDAO' 클래스는 여러 개의 메소드가 사용되기 때문에 각각의 메소드끼리 데이터베이스 접근에 있어서 서로 마찰이 일어나지 않도록 'PreparedStatement' 는 각각의 메소드 내에서 하나씩 추가하도록 하겠습니다

```

//작성일자 메소드
public String getDate() {
    String sql = "select now()";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            return rs.getString(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return ""; //데이터베이스 오류
}

```

'getDate()' - 작성일자를 구하는 메소드

글쓰기를 하는 현재 날짜와 시간을 출력해주는 메소드입니다

```

41 //게시글 번호 부여 메소드
42 public int getNext() {
43     //현재 게시글을 내림차순으로 조회하여 가장 마지막 글의 번호를 구한다
44     String sql = "select bbsID from bbs order by bbsID desc";
45     try {
46         PreparedStatement pstmt = conn.prepareStatement(sql);
47         rs = pstmt.executeQuery();
48         if(rs.next()) {
49             return rs.getInt(1) + 1;
50         }
51         return 1; //첫 번째 게시물인 경우
52     } catch (Exception e) {
53         e.printStackTrace();
54     }
55     return -1; //데이터베이스 오류
56 }
57

```

'getNext()' 메소드 - 새로운 게시글 번호 부여

게시글 번호를 부여하는 메소드입니다. 현재 저장 되어 있는 모든 게시글을 내림차순으로 정렬하여 가장 상위의 게시글번호를 구해서 그 게시글 번호에 1을 더한 값을 새로 작성되는 게시글의 새 번호로 부여합니다. 예를 들어 현재 'bbs.jsp' 게시판 메인 페이지에 총 13개의 글이 작성 되어 있다면, 가장 최신 글의 번호인 '13'에서 +1을 더한 값인 '14'가 새로 만들어지는 게시글의 번호로 지정이 됩니다.



//글쓰기 메소드

```
public int write(String bbsTitle, String userID, String bbsContent) {
    String sql = "insert into bbs values(?, ?, ?, ?, ?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, getNext());
        pstmt.setString(2, bbsTitle);
        pstmt.setString(3, userID);
        pstmt.setString(4, getDate());
        pstmt.setString(5, bbsContent);
        pstmt.setInt(6, 1); //글의 유효번호
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1; //데이터베이스 오류
}
```

'write' 메소드 - 글제목, 사용자ID, 글내용을 전달받아 DB에 새로운 데이터를 입력  
게시글 번호 -> 글 제목 -> 사용자ID -> 작성일자 -> 글 내용 -> 유효번호  
순서대로 데이터를 셋팅하고 데이터베이스로 전송합니다.

매개변수로 글의 제목, 글을 입력한 사용자 ID, 글 내용을 입력받고 실제 데이터베이스의  
테이블 속성 순서에 맞게 데이터를 입력하는 메소드입니다.

```
package bbs;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
public class BbsDAO{
```

```
    private Connection conn;
```

```
    private ResultSet rs;
```

```
//기본 생성자
```

```
public BbsDAO (){
```

```
    try {
```

```
        String dbURL = "jdbc:mariadb://localhost:3306/bbs";
```



```

        String dbID = "root";
        String dbPassword = "001025";
        Class.forName("org.mariadb.jdbc.Driver");
        conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
    }catch (Exception e) {
        e.printStackTrace();
    }
}

```

*//작성일자 메소드*

```

public String getDate (){
    String sql = "select now()";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            return rs.getString(1);
        }
    }catch (Exception e) {
        e.printStackTrace();
    }
    return ""; //데이터베이스 오류
}

```

*//게시글 번호 부여 메소드*

```

public int getNext (){
    //현재 게시글을 내림차순으로 조회하여 가장 마지막 글의 번호를 구한다
    String sql = "select bbsID from bbs order by bbsID desc";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            return rs.getInt(1) + 1;
        }
        return 1; //첫 번째 게시글인 경우
    }catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        return -1; //데이터베이스 오류
    }

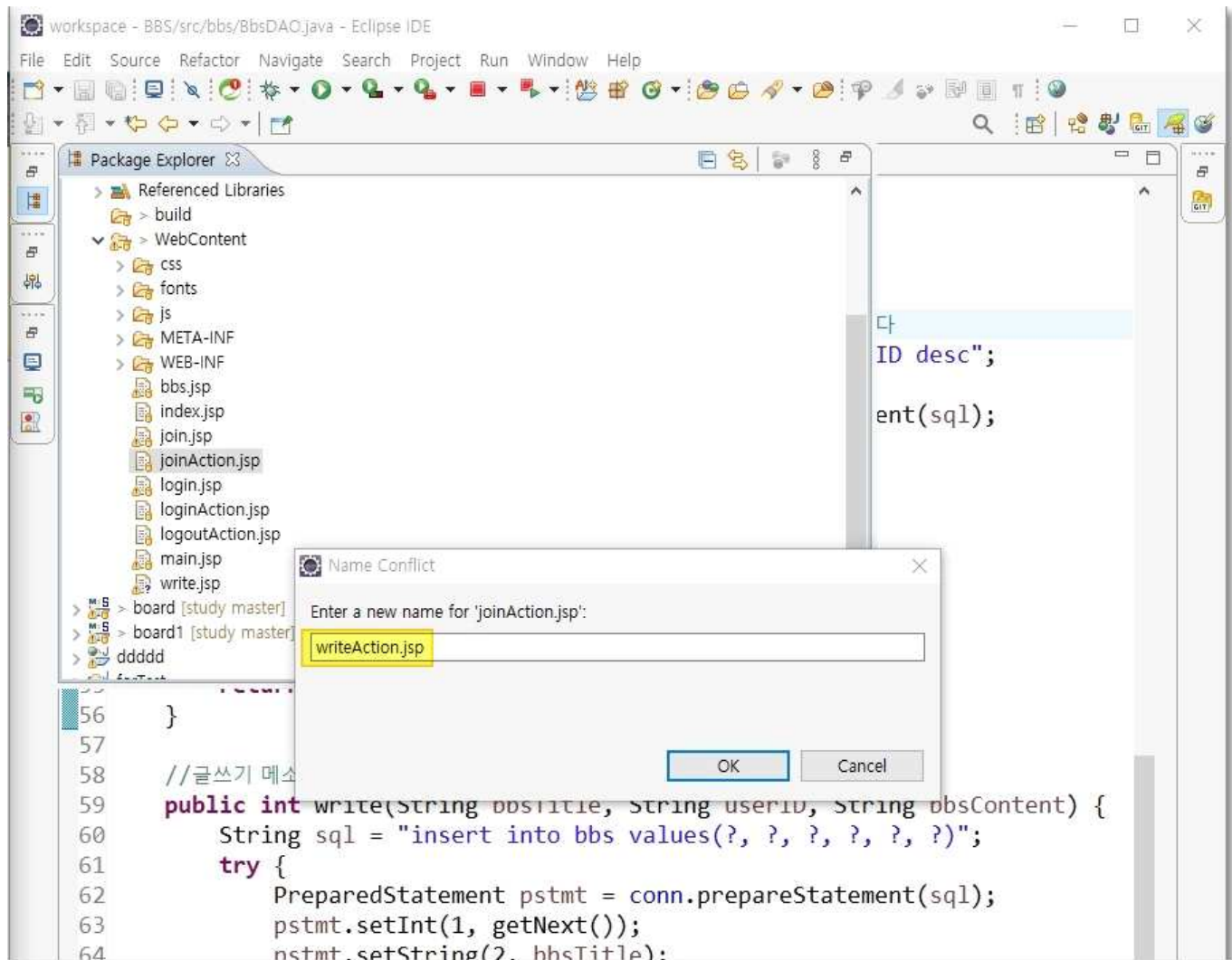
    //글쓰기 메소드
    public int write (String bbsTitle, String userID, String bbsContent){
        String sql = "insert into bbs values(?, ?, ?, ?, ?, ?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, getNext());
            pstmt.setString(2, bbsTitle);
            pstmt.setString(3, userID);
            pstmt.setString(4, getDate());
            pstmt.setString(5, bbsContent);
            pstmt.setInt(6, 1); //글의 유효번호
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1; //데이터베이스 오류
    }
}

```

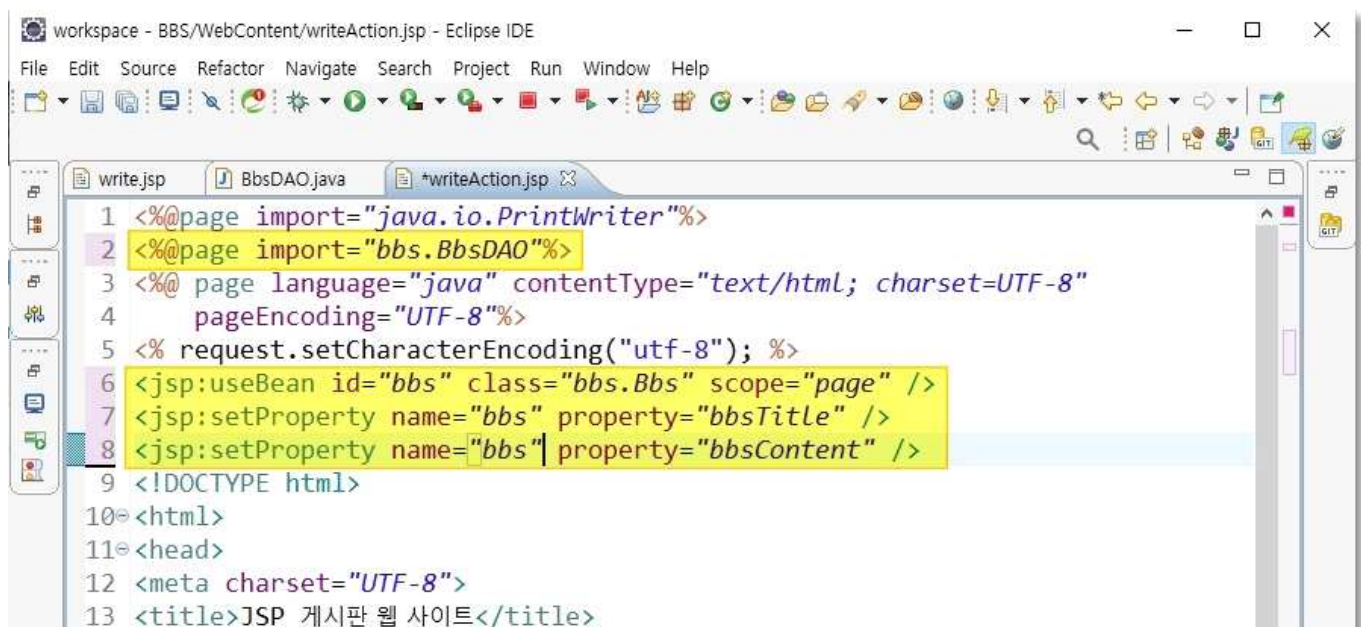
---

# writeAction.jsp

글쓰기 버튼을 눌렀을 때 실제로 게시글 내용이 데이터베이스에 입력이 되는 로직을 수행할 수 있도록 'JSP' 페이지를 만듭니다



'joinAction.jsp' 을 복사하여 'writeAction.jsp' 를 생성합니다



상단에 페이지 세팅을 수정합니다

```
<%
// 현재 세션 상태를 체크한다
String userID = null;
if(session.getAttribute("userID") != null){
    userID = (String)session.getAttribute("userID");
}
// 로그인한 사람만 글을 쓸 수 있도록 코드를 수정한다
if(userID == null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('로그인을 하세요')");
    script.println("location.href='login.jsp'");
    script.println("</script>");
}else{
    // 입력이 안 된 부분이 있는지 체크한다
    if(bbs.getBbsTitle() == null || bbs.getBbsContent() == null){
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('입력이 안 된 사항이 있습니다')");
        script.println("history.back()");
        script.println("</script>");
    }else{
        // 정상적으로 입력이 되었다면 글쓰기 로직을 수행한다
        BbsDAO bbsDAO = new BbsDAO();
        int result = bbsDAO.write(bbs.getBbsTitle(), userID, bbs.getBbsContent());
        // 데이터베이스 오류인 경우
        if(result == -1){
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글쓰기에 실패했습니다')");
            script.println("history.back()");
            script.println("</script>");
            // 글쓰기가 정상적으로 실행되면 알림창을 띄우고 게시판 메인으로 이동한다
        }else {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글쓰기 성공')");
            script.println("location.href='bbs.jsp'");
            script.println("</script>");
        }
    }
}
}
```

1. 먼저 로그인을 한 사람만이 글을 쓸 수 있도록 코드를 수정합니다
2. 글 제목과 글 내용 중 누락된 사항이 있는지 체크합니다
3. 로그인을 하였고 모든 내용이 정상적으로 입력이 되었을 때 '글쓰기 성공' 알림창을 띄우고 게시판 메인 페이지('bbs.jsp')로 이동할 수 있도록 코드를 작성합니다

```
<%@page import="java.io.PrintWriter"%>
```

```
<%@page import="bbs.BbsDAO"%>
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<% request.setCharacterEncoding("utf-8"); %>
<jsp:useBean id = "bbs" class = "bbs.Bbs" scope = "page" />
<jsp:setProperty name = "bbs" property = "bbsTitle" />
<jsp:setProperty name = "bbs" property = "bbsContent" />
<!DOCTYPE html >
<html >
<head >
<meta charset = "UTF-8">
<title >JSP 게시판 웹 사이트 </title >
</head >
<body >
    <%
        // 현재 세션 상태를 체크한다
        String userID = null;
        if(session.getAttribute("userID") != null){
            userID = (String)session.getAttribute("userID");
        }
        // 로그인한 사람만 글을 쓸 수 있도록 코드를 수정한다
        if(userID == null){
            PrintWriter script = response.getWriter();
            script.println("<script >");
            script.println("alert('로그인을 하세요')");
            script.println("location.href='login.jsp'");
            script.println("</script >");
        }else{
            // 입력이 안 된 부분이 있는지 체크한다
            if(bbs.getBbsTitle() == null || bbs.getBbsContent() == null){
                PrintWriter script = response.getWriter();
                script.println("<script >");
                script.println("alert('입력이 안 된 사항이 있습니다')");
                script.println("history.back()");
                script.println("</script >");
            }else{
                // 정상적으로 입력이 되었다면 글쓰기 로직을 수행한다
                BbsDAO bbsDAO = new BbsDAO();
                int result = bbsDAO.write(bbs.getBbsTitle(), userID,

```

```

bbs.getBbsContent());

// 데이터베이스 오류인 경우
if(result == -1){
    PrintWriter script = response.getWriter();
    script.println("<script >");
    script.println("alert('글쓰기에 실패했습니다')");
    script.println("history.back()");
    script.println("</script >");
    // 글쓰기가 정상적으로 실행되면 알림창을 띄우고 게시판 메인
    으로 이동한다

}
else {
    PrintWriter script = response.getWriter();
    script.println("<script >");
    script.println("alert('글쓰기 성공')");
    script.println("location.href='bbs.jsp'");
    script.println("</script >");
}
}

%>
</body >
</html >

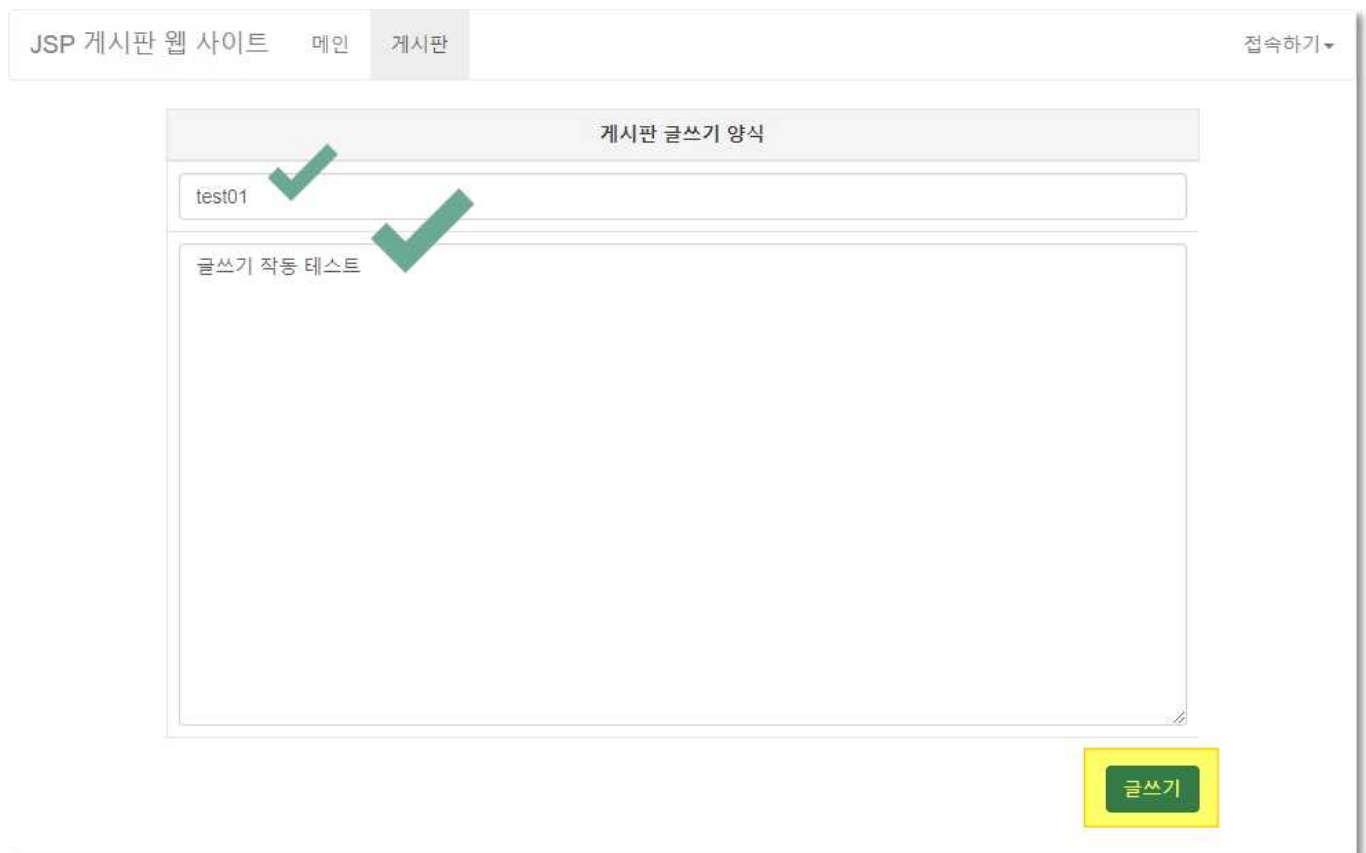
```

---

# 글쓰기 기능 실행 테스트

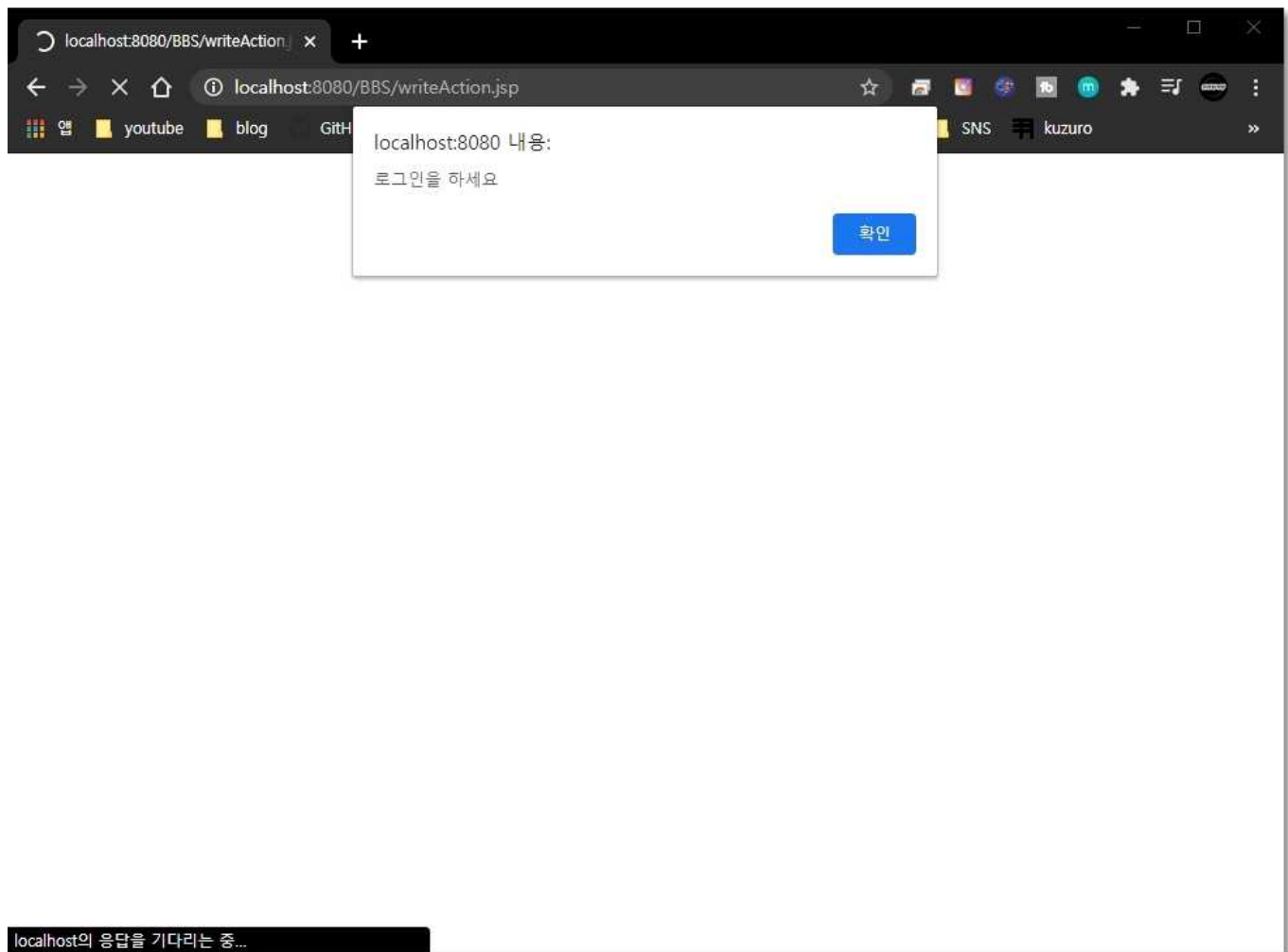


[index.jsp 실행] - [게시판] - [글쓰기]

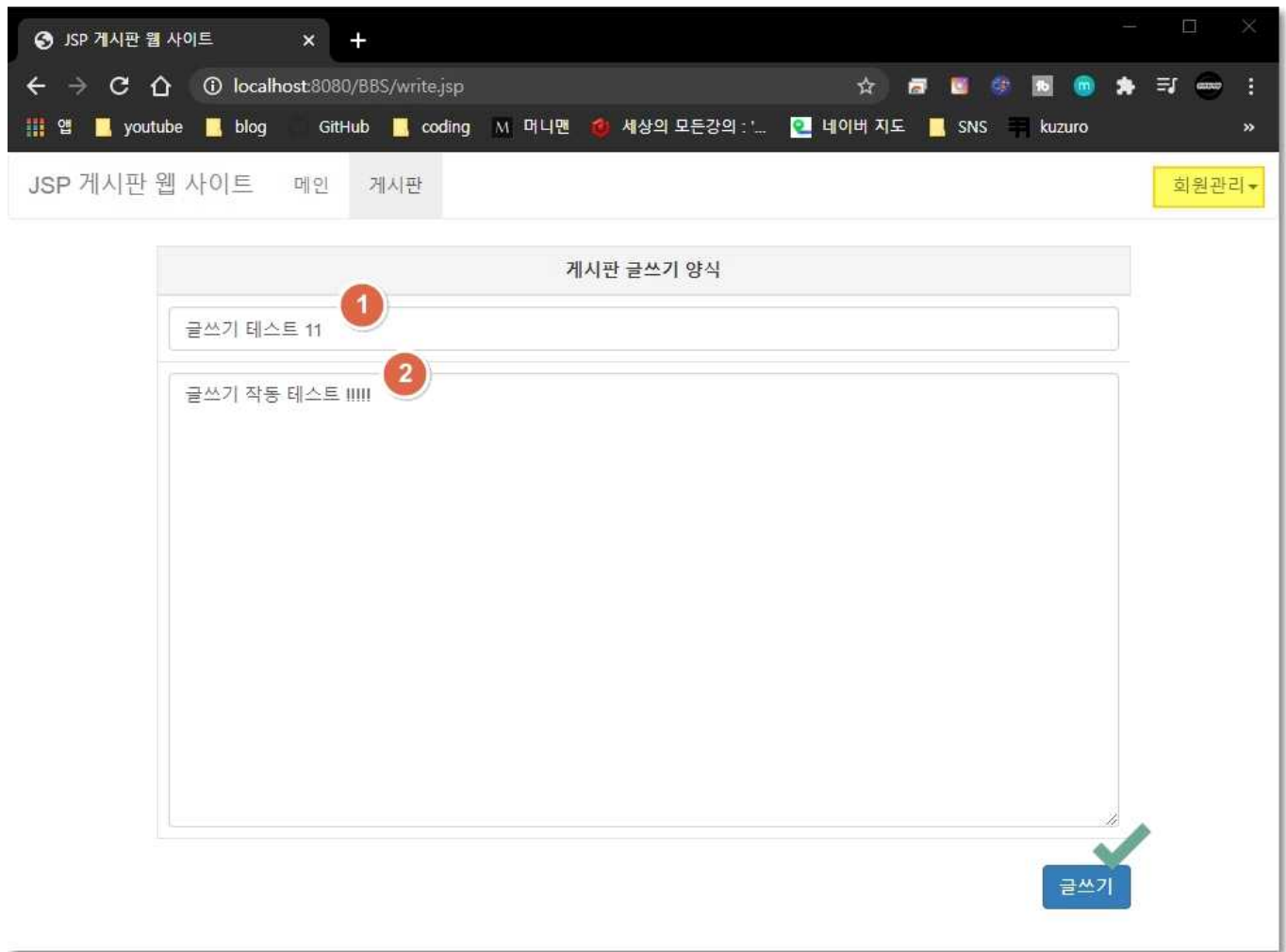


글 제목과 글 내용을 입력 후 '글쓰기' 버튼 누르기

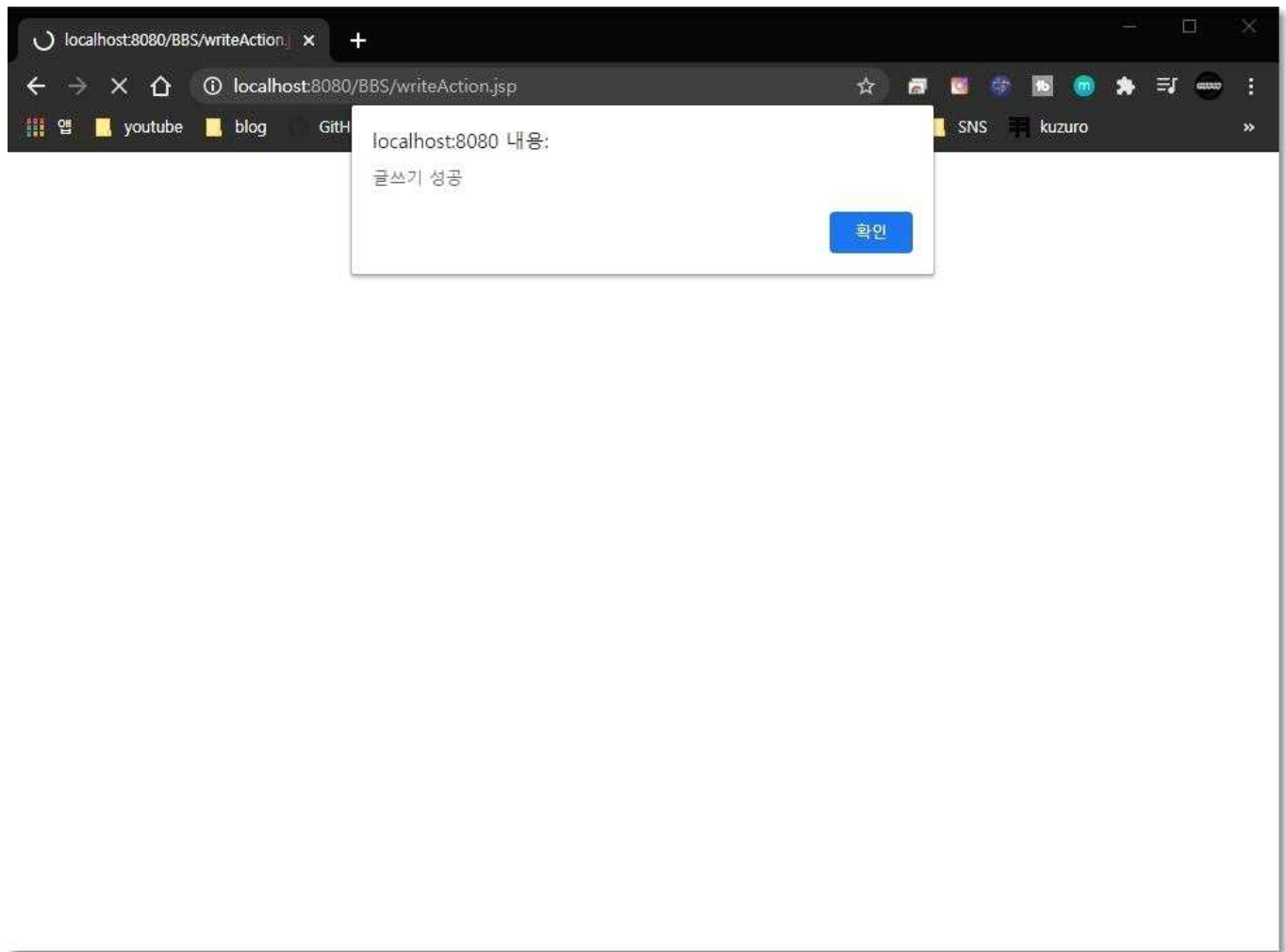




로그인을 하지 않아서 글쓰기에 실패하고 로그인을 하라는 알림창이 뜹니다  
확인을 누르면 바로 로그인 창으로 이동할 겁니다  
작성한 코드가 정상적으로 작동이 되는 것을 확인할 수 있습니다



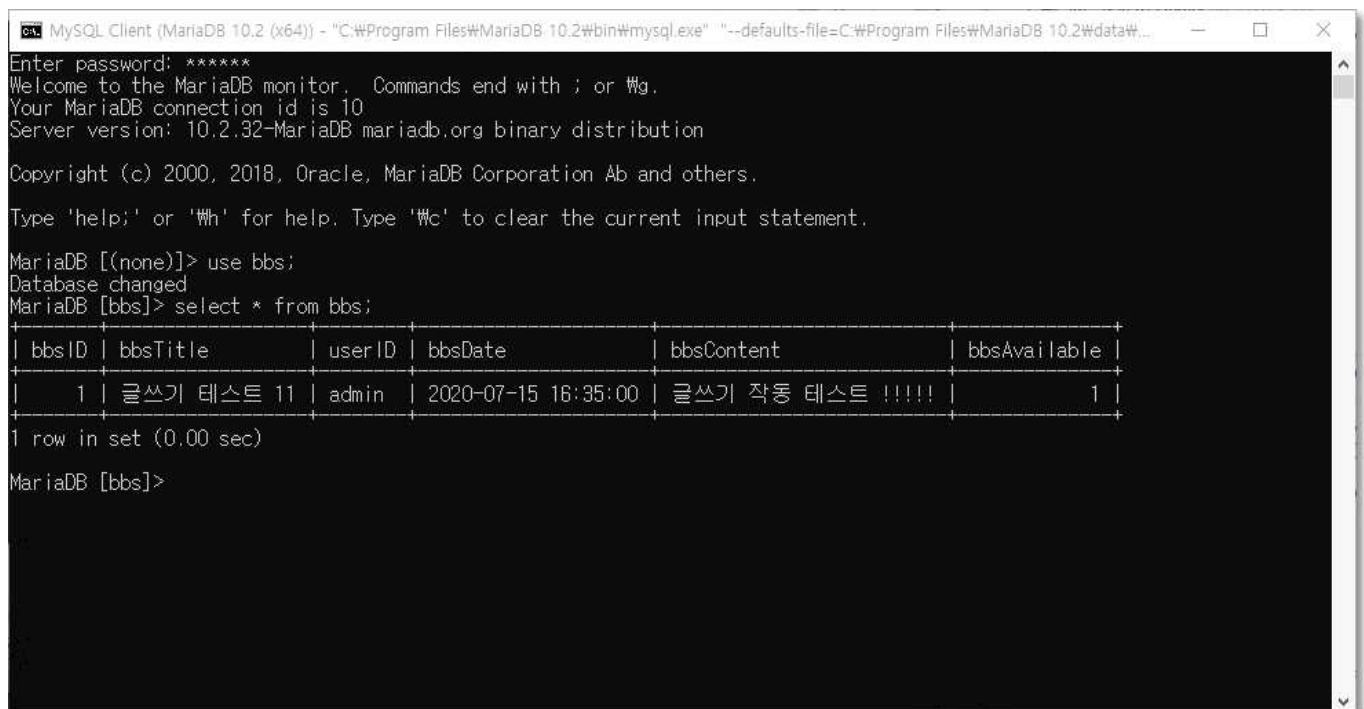
현재는 로그인한 상태이고 우측상단에 메뉴도 '회원관리'로 바뀐 것을 확인할 수 있습니다. 이제 로그인을 하였으니 다시 글 제목과 글 내용을 입력한 후 '글쓰기' 버튼을 눌러 보겠습니다.



글쓰기가 정상적으로 실행되고 알림창까지 뜨는 것을 볼 수 있습니다



글쓰기에는 성공하였지만 아직 '게시글 목록' 기능을 구현하지 않았습니다  
따라서 게시판 메인 페이지는 이전과 똑같은 것을 볼 수 있습니다



하지만 데이터베이스에 접속해보면 '글쓰기' 가 정상적으로 실행되어 데이터가 들어가 있는

것을 확인할 수 있습니다