**보고서**

**DB와 연계한 회원관리 페이지 만들기**



|  |  |
| --- | --- |
| **제출일자** | 2022년 05월 07일 (토) |
| **교과목** | 클라우드시스템 |
| **담당교수** | 박유현 교수님 |
| **학과** | 컴퓨터소프트웨어공학과 |
| **이름(학번)** | 박상현 (20183145) |
| **학년** | 3학년 |
| **분반** | 1분반 |

|  |  |
| --- | --- |
| **1** | **개요** |

**◎ 프로그램 동작 내용**

1. **Index.html**
   * 각 button의 onclick 속성을 이용해 ‘로그인’과 ‘회원가입’을 분리한다
   * window.location 메소드를 사용하여 url을 직접 연결한다
2. **SignIn.jsp**
   * MySQL과 연동을 수행한다
   * 데이터베이스에 있는 값은 가변길이 이므로, ArrayList를 사용한다
   * 데이터베이스의 데이터와 입력한 데이터를 비교한다
3. **SignUp.jsp**
   * MySQL과 연동을 수행한다
   * stmt.executeUpdate 메소드를 사용하여 입력한 데이터를 데이터베이스에 insert 한다

|  |  |
| --- | --- |
| **2** | **소스 설명** |

|  |
| --- |
| **Index.html** |
| <html>  <title>Login\_20183145 박상현</title>  <head>  <meta charset="utf-8">  </head>  <body>  <p><h2>Login</h2></p>  <p> ID <input type="text" id="id" placeholder="Input ID" required> </p>  <p> P/W <input type="password" id="pw" placeholder="Input P/W"> </p>  <!-- 2개의 button 사용 -->  <p> **<input type="button" value="Sign in" onclick="signInClick()">**  **<input type="button" value="Sign up" onclick="signUpClick()">**</p>  </body>    **<script type="text/javascript">**  function signInClick() {  //input 값 가져오기  **const id = document.getElementById('id').value;**  const pw = document.getElementById('pw').value;    //jsp 파일의 url 이동  **window.location='http://101.xxx.xxx.xxx:80/SignIn.jsp?id='+id+'&pw='+pw;**  }  function signUpClick() {  const id = document.getElementById('id').value;  const pw = document.getElementById('pw').value;  window.location='http://101.xxx.xxx.xxx:80/SignUp.jsp?id='+id+'&pw='+pw;  }  **</script>**  </html> |

* **<input type="button" value="Sign in" onclick="signInClick()">**
  + **onclick 속성 :** button을 클릭 시 해당 함수를 실행한다 //JS 속성
* **<script type="text/javascript"> </script>**
  + Javascript 구문을 수행한다
* **const VALUE = document.getElementById(id).value**
  + id의 문자열을 반환한다
* **window.location=STR**
  + STR인 URL에 이동한다

|  |
| --- |
| **SignIn.jsp** |
| <%@ page contentType ="text/html; charset=utf-8" %>  **<%@ page import “=java.sql.\*" %>**  **<%@ page import="java.util.ArrayList" %>**  <%  request.setCharacterEncoding("utf-8");  //url 쿼리스트링을 통한 입력 값 저장  **String id = request.getParameter("id");**  String pw = request.getParameter("pw");  //DB에 저장된 데이터  **ArrayList <String> passId = new ArrayList<String>();**  ArrayList <String> passPw = new ArrayList<String>();  boolean Enter = false;  // MySQL JDBC Driver Loading  **Class.forName("com.mysql.cj.jdbc.Driver"); //JDBC 로딩 함수**  **String jdbcDriver ="jdbc:mysql://localhost:3306/TestDB?serverTimezone=UTC";**  String dbUser ="tester"; //mysql id  String dbPass ="1234"; //mysql password  String query ="select \* from user"; //query에 user의 \* 속성 출력  // Create DB Connection  **Connection conn = DriverManager.getConnection(jdbcDriver, dbUser, dbPass); //접속 시**도  // Create Statement  **Statement stmt = conn.createStatement(); //접속한 객체에 Statement 생성**  // Run Qeury  **ResultSet rs = stmt.executeQuery(query); //query 실행**  //rs : query를 실행한 결과 result set  // Print Result (Run by Query)    **while(rs.next()) { //rs를 한 줄씩 반복**  passId.add(rs.getString("id"));  passPw.add(rs.getString("passwd"));  } //while(rs.next()) 끝  %>  <%  **for(int i=0; i<passId.size(); i++){**  **if(id.equals(passId.get(i))&&pw.equals(passPw.get(i))){ //ID와 PW가 일치할 시**  Enter = true;  break; //일치하면 바로 for문을 빠져 나간다  }  else{  Enter = false;  }  }  if(Enter == true)  out.println("Login Succese!");  else  out.println("Login fail");  %> |

* **<%@ page import ="java.sql.\*" %>**
  + java에서 SQL을 사용하기 위한 import 선언
* **<%@ page import="java.util.ArrayList" %>**
  + java에서 ArrayList를 사용하기 위한 import 선언
* **Class.forName("com.mysql.cj.jdbc.Driver")**
  + JDBC 드라이버 로딩
* **String jdbcDriver ="URL"**
  + DB에 접속할 URL
* **Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)**
  + 접속을 위한 Connection 객체 생성
    - **URL** : 접속할 DB 주소
    - **USER** : 사용자의 이름
    - **PASSWORD** : 사용자의 패스워드
* **Statement stmt = conn.createStatement()**
  + SQL 명령문을 전달하기 위한 Statement 객체 생성
* **ResultSet rs = stmt.executeQuery(query)**
  + Statement 객체에 query 실행
* **while(rs.next()) { } :** DB 데이터를 하나씩 반복하며 검색한다

|  |
| --- |
| **SignUp.jsp** |
| <%@ page contentType ="text/html; charset=utf-8" %>  <%@ page import “=java.sql.\*" %>  <%  request.setCharacterEncoding("utf-8");  String id = request.getParameter("id");  String pw = request.getParameter("pw");  boolean Enter = false;  // MySQL JDBC Driver Loading  Class.forName("com.mysql.cj.jdbc.Driver"); //JDBC 로딩 함수  Connection conn =null;  Statement stmt =null;  ResultSet rs =null;  String jdbcDriver ="jdbc:mysql://localhost:3306/TestDB?serverTimezone=UTC";  String dbUser ="tester"; //mysql id  String dbPass ="1234"; //mysql password  String query ="select \* from user"; //query에 user의 \* 속성 출력  // Create DB Connection  conn = DriverManager.getConnection(jdbcDriver, dbUser, dbPass); //접속 시도  // Create Statement  stmt = conn.createStatement(); //접속한 객체에 Statement 생성  // Run Qeury  rs = stmt.executeQuery(query); //query 실행  //rs : query를 실행한 결과 result set  //SQL 삽입  **String insert = "INSERT INTO user(ID, passWd) VALUES ('" + id + "', '" + pw + "')";**  //executeUpdate : SQL의 DML 수행  **int count = stmt.executeUpdate(insert);**  if(count == 1){ //성공적으로 저장 시  **out.println("<script>alert('회원가입 완료');</script>");**  }  else{  out.println("<script>alert('회원가입 실패');</script>");  }  stmt.close();  conn.close();  %> |

* Int stmt.executeUpdate(STR) SQL의 DML(STR) 수행
  + return num : num만큼 반영된 레코드의 건수
  + return -1 : create / drop 수행

//DML Query 문은 따로 변수(String)로 선언해 주는 것이 좋다

* **out.println("<script>alert('STR');</script>")**
  + jsp에서 내용을 STR로 하는 알람(alert)를 보내는 구문

|  |  |
| --- | --- |
| **3** | **실행 결과** |

|  |
| --- |
| **Login 페이지 (Index.html)** |
|  |

|  |
| --- |
| **로그인 성공 페이지 (SignIn.jsp)** |
|  |

|  |
| --- |
| **로그인 실패 페이지 (SignIn.jsp)** |
|  |

|  |
| --- |
| **회원가입 성공 페이지 (SignUp.jsp)** |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |

|  |
| --- |
| **DB User 테이블** |
| +------+--------+  | ID | passwd |  +------+--------+  | deu | 1234 |  | root | 0000 |  | test | 123456 |  +------+--------+ |

|  |  |
| --- | --- |
| 4 | **총평** |

지금까지 알고 있었던 submit 방식은 버튼이 하나일 때만 가능하였습니다. 이에 관련해서 교수님께 질문도 해 보았고, 인터넷 검색을 많이 했습니다. JavaScript를 사용하는 방식도 있었지만, button을 두 개 만들어 각각 다른 함수를 호출하는 방식을 사용하였습니다.

하지만 해당 방식을 사용하면 URL에 쿼리스트링이 그대로 노출되어 POST 방식이 아닌 GET 방식이 사용된다는 단점이 존재합니다. 이에 대해서는 조금 더 생각이 필요할 거 같습니다.

이번 과제를 통해 기말 프로젝트를 시작할 수 있을 것으로 보입니다. 웹사이트의 기초를 뗀 것 같아 앞으로의 활용이 더욱 기대됩니다.