

## KD2기 Web Backend 역량 평가

2023. 06. 23.

이름 :

### [프로그램 개요]

1. 이 프로젝트는 Java, Spring, Mybatis를 이용한 Web Application 회원으로 가입된 사용자가 자신의 todo list를 웹으로 관리하는 프로그램이다.

제공되는 압축파일을 임의로 생성한 작업폴더에서 압축 해제하고 코드를 분석한 후 미완성된 부분의 코드를 완성한다

**모든 코드를 확인하여 // Code here** 라고 되어있는 부분을 완성해야 하며 **단위기능이 실행되어야 점수를 획득**할 수 있다.

2. 완성된 모든 \*.java 파일은 아래와 같이 Export하여 압축된 파일을 제출한다.

**제출파일명: KD2기\_홍길동.zip**

### [Database 생성]

1. 아래와 같은 조건의 테이블과 시퀀스를 생성하고, 그 결과를 todolist.sql에 저장하시오.

(테이블을 생성하면서 제약조건을 부여할 것.)

1) person: 회원의 정보를 저장하기 위한 테이블

컬럼명	컬럼타입	제약조건	설명
todoid	가변길이 50	PRIMARY KEY	아이디
todopwd	가변길이 100	NOT NULL	비밀번호
todoname	가변길이 20	NOT NULL	회원의 이름
enabled	정수 1자리	default 1	계정상태. 1:사용가능, 0:사용불가능
rolename	가변길이 20	default 'ROLE_USER'	사용자 권한. 모두 'ROLE_USER'로 처리

2) todolist : todo 정보를 저장하기 위한 테이블

컬럼명	컬럼타입	제약조건	설명
seqno	정수형	PRIMARY KEY	일련번호
todoid	가변길이 50	FOREIGN KEY	person 테이블의 todoid컬럼을 외래키로 참조 부모테이블의 컬럼이 삭제되면 todolist 테이블의 데이터도 함께 삭제되도록 <b>delete cascading 설정할 것</b>
regdate	날짜	기본값 SYSDATE	등록 날짜
status	가변길이 10	CHECK (‘완료’, ‘진행’, ‘지연’)	진행상태(아래 3가지 상태 값을 갖는다.) (‘완료’, ‘진행’, ‘지연’)
importance	가변길이 10	CHECK (‘높음’, ‘보통’, ‘낮음’)	중요도 (아래 3가지 상태 값을 갖는다.) (‘높음’, ‘보통’, ‘낮음’)
categories	가변길이 6	기본값 ‘개인’ CHECK (‘개인’, ‘회사’)	todo 종류
todo	가변길이 3000	기본값 ‘할일 없음’	- 할일 정보

3) todolist\_seq : todo 테이블의 seqno 컬럼의 일련번호를 위한 시퀀스 객체

## [기능 별 요구사항]

### 1) 메인

- 프로그램이 구동되면 아래와 같은 화면이 출력된다.
- 로그인 전

[ TodoList ]

[회원가입](#)  
[로그인](#)

- 로그인 후


[ TodoList ]

hong님, [로그아웃](#)  
[TodoList](#)

### 2) 회원가입

- 아이디, 비밀번호, 이름을 입력 받아 DB에 저장한다.


[ 회원 가입 ]



아이디	<input type="text" value="hong"/>	사용가능한 아이디입니다.
비밀번호	<input type="password" value="...."/>	
이름	<input type="text" value="홍길동"/>	

- 아이디는 3~5자 이내로 입력 받는다.
- 아이디는 중복여부를 **ajax 기능을 이용**하여 검증한다.
- 아이디가 중복될 경우 “사용할 수 없는 아이디입니다”라는 메시지를 붉은 색으로 나타나도록 한다.

[ 회원 가입 ]



아이디	<input type="text" value="ho"/>	아이디는 3~5자 이내로 작성해주세요
비밀번호	<input type="password" value="...."/>	
이름	<input type="text" value="홍길동"/>	

- 비밀번호는 3~5자 이내로 입력 받는다.
- 이름은 반드시 입력 받는다.
- “가입” 버튼을 클릭했을 때 아이디와 비밀번호와 이름을 재검증한다.

### 3) 로그인 / 로그아웃

- 로그인 화면에서 아이디와 비밀번호를 입력하면 첫 화면으로 이동한다.

## [ 로그인 ]

<b>아이디</b>	<input type="text" value="아이디를 3~5자로 이내로 입력"/>
<b>비밀번호</b>	<input type="password" value="비밀번호를 3~5자 이내로 입력"/>

## [ 로그인 ]

<b>아이디</b>	<input type="text" value="aa"/>	아이디는 3~5자 이내로 작성해주세요
<b>비밀번호</b>	<input type="password" value=".."/>	비밀번호는 3~5자 이내로 작성해주세요

- 아이디/비번 검증 : 아이디와 이메일은 3~5자 이내로 입력하지 않을 경우 각 품의 오른쪽에 붉은 색으로 올바른 입력이 아님을 나타내고
- 로그인 버튼을 클릭했을 때 올바른 아이디와 비번이 입력되지 않을 경우 경고창을 출력한다.

localhost:9991 내용:

로그인을 위해 데이터를 정확히 입력하세요

### 4) todo 등록

- 첫 화면에서 로그인인 된 후 todoList를 클릭하면 아래와 같이 로그인한 아이디, 작성일, 상태, 중요도, 분류, to Do를 입력할 수 있는 화면이 출력되고, 입력이 끝난 후 “할일 등록” 버튼을 클릭하면 해당 내용을 ajax로 DB에 저장한 후 내용이 화면에 출력되도록 한다.

## [ Todo List ]

<b>아이디</b>	hong
<b>작성일</b>	2023-06-20 <input type="button" value="📅"/>
<b>상태</b>	<input checked="" type="radio"/> 진행 <input type="radio"/> 지연 <input type="radio"/> 완료
<b>중요도</b>	높음 ▼
<b>분류</b>	<input type="radio"/> 개인 <input checked="" type="radio"/> 회사
<b>To Do</b>	<div style="border: 1px solid #ccc; min-height: 40px; padding: 5px;">프로젝트 회의 및 제안 분석</div>

[ Todo List ]

아이디	hong
작성일	2023-06-20
상태	<input checked="" type="radio"/> 진행 <input type="radio"/> 지연 <input type="radio"/> 완료
중요도	높음 ▼
분류	<input type="radio"/> 개인 <input checked="" type="radio"/> 회사
To Do	<div style="border: 1px solid #ccc; padding: 5px; min-height: 40px;"> 프로젝트 회의 및 제안 분석 </div>

할일 등록
취소

(todo 보기)

날짜	분류	상태	중요도	해야 할 일	비고
2023-06-23	회사	진행	높음	KD2기 웹 프로젝트 역량평가	<span style="border: 1px solid #ccc; padding: 2px 5px;">삭제</span>
2023-06-19	개인	진행	높음	KD2기 특강	<span style="border: 1px solid #ccc; padding: 2px 5px;">삭제</span>
2023-06-12	개인	진행	높음	ajax를 이용하여 아이디와 비번 확인작업	<span style="border: 1px solid #ccc; padding: 2px 5px;">삭제</span>

## 5) todo 삭제

- todo list 각 항목에서 **삭제** 버튼 클릭하면 삭제 여부를 아래와 같이 삭제 여부를 묻는 창을 띄우고 확인버튼을 클릭하면 해당 todo를 DB에서 삭제한다.
- 삭제 완료 후에는 후 ajax기능을 이용하여 todo를 다시 출력한다.

localhost:9991 내용:

정말로 삭제하시겠습니까?

확인
취소

## 6) 로고 아이콘 클릭 시 첫 화면으로 이동

※ 프로젝트는 하나의 단위 별로 실행 가능해야 하며 코드가 부분적으로 완성되어 있다 하더라도 해당 단위가 실행이 안될 경우 감점한다. (예를 들어 입력일 경우 입력에 따른 모든 기능이 실행되어야 한다.)

## [평가 기준]

No.	평가항목	배점
1	<p><b>쿼리문 생성</b> (생성한 쿼리문은 프로젝트 내의 저장할 것)</p> <ul style="list-style-type: none"> <li>- personal 테이블 생성 <b>(5점)</b></li> <li>- todoist 테이블 생성 <b>(5점)</b></li> <li>- todoist_seq 시퀀스 객체 생성 <b>(2점)</b></li> </ul> <p>컬럼타입이나 제약조건을 잘못 생성했을 경우 0점 처리 쿼리문의 실행 문을 todoist.sql에 저장하지 않았을 경우 0점 처리</p>	12점
2	<p><b>첫 화면 (index.html)</b></p> <ul style="list-style-type: none"> <li>- 로그인한 사람과 로그인하지 않은 사람에게 출력되는 메뉴가 달라야 한다. <b>(5점)</b></li> <li>- 첫 화면을 제외한 모든 html문서에서 로고를 클릭하면 첫 화면으로 이동 <b>(5점)</b></li> </ul>	10점
3	<p><b>회원가입 기능</b></p> <p>1) 화면(html): jQuery 기능을 사용할 때는 주어진 파일에 코딩할 것</p> <ul style="list-style-type: none"> <li>- 아이디는 ajax 기능을 이용하여 중복체크 <b>(5점)</b></li> <li>- 가입버튼 클릭 시 비밀번호 3자리, 이름 입력 여부체크하고, 아이디상태를 다시 체크하여 문제가 없을 때 가입처리 완료 <b>(5점)</b></li> </ul> <p>2) 비밀번호는 Service 파일에서 Security에서 제공하는 암호화 기능을 이용 저장 <b>(3점)</b></p> <p>3) Mapper XML 내의 쿼리문</p> <ul style="list-style-type: none"> <li>- 아이디 중복체크 <b>(2점)</b></li> <li>- 회원가입 <b>(3점)</b></li> </ul>	18점
4	<p><b>Security 설정 (WebSecurityConfig.java)</b></p> <ul style="list-style-type: none"> <li>- 이미지, 자바스크립트, CSS와 인증 받지 않은 사용자의 접근여부 설정 <b>(2점)</b> 로그인, 로그아웃, 회원가입, 아이디 체크, 이미지, CSS, JS는 누구나 접근가능</li> <li>- 로그인을 위한 쿼리 <b>(1.5점)</b></li> <li>- 권한 확인을 확인한 쿼리 <b>(1.5점)</b></li> </ul>	5점
4	<p><b>Todo 기능</b></p> <p>1) todo 입력화면 작성 <b>(10점)</b></p> <ul style="list-style-type: none"> <li>- 아이디: 로그인한 회원의 아이디가 출력</li> <li>- 상태: 진행, 지연, 완료 3가지 선택을 할 수 있는 라디오 폼. “진행”이 선택되어 있다.</li> <li>- 중요도: 높음, 보통, 낮음 3가지 선택할 수 있는 선택 폼. “높음”이 선택되어 있다.</li> <li>- To Do: 할 일 정보를 입력할 수 있는 Textarea 폼.</li> </ul> <p>2) todo 등록 <b>(총 15점)</b></p> <ul style="list-style-type: none"> <li>- “할일 등록” 버튼을 클릭하면 입력된 정보를 ajax 기능을 이용하여 DB에 저장한다.</li> <li>- TodoListServiceImpl.java 파일은 TodoListService.java 인터페이스를 구현하여 이에 맞춰 작성할 것!</li> <li>- 쿼리 구현: 사용자가 뷰단에서 날짜를 선택하지 않고 등록할 경우 동적쿼리를 이용하여 오늘 날짜로 입력되도록 한다. <b>(10점)</b></li> </ul>	50점

	<ul style="list-style-type: none"> <li>- 등록이 완료된 후 js의 cleanup() 함수를 호출하여 입력화면을 깨끗이 초기화 한다. (5점)</li> </ul> <p>3) todo 출력 (총 10점)</p> <ul style="list-style-type: none"> <li>- 할 일을 등록하면 등록된 todolist의 내용을 날짜 별 역순으로 조회한 후 화면에 출력</li> <li>- 화면 구현: (5점)</li> <li>- 쿼리 구현: (5점)</li> </ul> <p>4) todo 삭제 (총 15점)</p> <ul style="list-style-type: none"> <li>- todo 화면에서 삭제 버튼을 클릭하면 삭제여부를 묻는 창 출력. (5점)</li> <li>- 확인을 클릭하면 DB에서 삭제된다. (쿼리구현 (5점))</li> <li>- 등록이나 삭제가 완료되면 ajax를 이용하여 화면에서 결과 재 출력하는 함수 (5점)</li> </ul>	
5	<b>Domain</b> <ul style="list-style-type: none"> <li>- Person.java: 개인 정보를 저장할 수 있는 파일(3점)</li> <li>- Todo.java: 할 일 정보를 저장할 수 있는 파일(2점)</li> </ul>	5점
총점		100점