Web 역량평가

**이름:**

1. 제출 형식

* 제공된 프로젝트를 STS로 import하여 완성한다.
* DB의 테이블 및 시퀀스 생성 쿼리를 프로젝트 내에 blog.sql 파일로 저장한다.
* 프로젝트 폴더를 압축하여 본인의 이름으로 파일명을 지정한다. (예: 홍길동.zip)
* 압축파일을 메일에 첨부하여 [sypark@softsociety.net](mailto:sypark@softsociety.net) 로 제출한다.

1. 프로그램의 조건

* 이 프로그램은 인터넷의 개인 블로그 서비스를 제공하는 WEB Application이다.
* Context root 경로는 “blog”로 지정한다. (접속 예: <http://localhost:8888/blog>)
* 가입한 사용자는 개인 블로그가 생성되며, 로그인을 하고 글을 작성, 삭제, 수정할 수 있다.
* 로그인하지 않은 사용자는 블로그 목록과 글 목록, 글 내용을 볼 수 있고, 추천 기능을 사용할 수 있다.
* 주어진 조건대로 Oracle에 테이블과 시퀀스를 생성하여 사용한다.
* 그 외의 부분은 제공된 코드의 주석을 참고하여 완성한다.

1. DB 테이블 구조
   1. **회원 정보 테이블 (테이블명 : blog\_member)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 타입 | 크기 | 제약조건 | 설명 |
| ID | 문자열 | 20 | 기본키 | 회원 ID |
| PASSWORD | 문자열 | 20 | NULL값 허용하지 않음 | 비밀번호 |
| NAME | 문자열 | 20 | NULL값 허용하지 않음 | 회원 이름 |
| PHOTO | 숫자 | 1 | 기본값 : 1 | 선택한 프로필 사진 번호  (1~4중에서 선택) |

* 1. **블로그 글 정보 테이블 (테이블명 : blog\_board)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 타입 | 크기 | 제약조건 | 설명 |
| BOARDNUM | 숫자 | 최대 | 기본키 | 회원 ID |
| ID | 문자열 | 20 | 외래키. | blog\_member 테이블의 ID 컬럼을 참조 |
| TITLE | 문자열 | 200 | NULL값 허용하지 않음 | 글 제목 |
| CONTENT | 문자열 | 2000 | NULL값 허용하지 않음 | 글 내용 |
| INPUTDATE | 날짜 |  | 기본값 : 시스템 현재 시간 | 작성 날짜 |
| LIKECNT | 숫자 | 최대 | 기본값 : 0 | 해당 글에 대한 추천수 |

* 1. **블로그 글의 일련번호에 사용할 시퀀스 객체 (이름: blog\_board\_seq)**

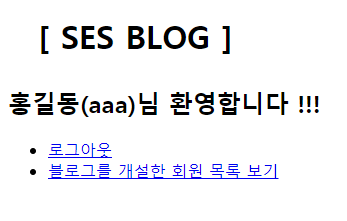
1. 프로그램의 구조
   1. 설정 파일 – 다음 파일들을 추가, 수정하여 사용한다.

* pom.xml – 프로젝트의 기본 버전 또는 라이브러리를 관리하는 Maven의 설정파일
* web.xml – WEB Application의 기본 설정 파일
* root-context.xml, servlet-context.xml : Spring의 설정 파일
* Log4j.xml : Logger 출력 수준 제어 설정 파일
* db.properties : DB 접속 정보를 가지고 있는 파일
* mybatis-config.xml : Mybatis의 환경 설정 파일
* memberMapper.xml / boardMapper.xml : SQL 구문을 정의한 파일
  1. VO 클래스
* MemberVO : 회원 정보를 저장하는 클래스. blog\_member 테이블의 컬럼과 매핑된다.
* BoardVO : 블로그의 글 정보를 저장하는 클래스. blog\_board 테이블의 컬럼과 매핑된다.
  1. Controller 클래스
* HomeController : 메인 화면으로 이동하는 콘트롤러
* MemberController : 회원 관련 기능 (가입, 로그인, 로그아웃, 회원목록)을 담당하는 콘트롤러
* BoardController : 블로그 관련 기능(글쓰기, 수정, 삭제, 목록보기, 추천하기)을 담당하는 콘트롤러.
  1. DAO, Mapper 클래스
* MemberDAO, BoardDAO : 회원, 블로그 관련 DB처리를 담당하는 클래스
* MemberMapper, BoardMapper : Mapper XML 파일과 매핑되는 인터페이스
  1. 기타 클래스
* LoginInterceptor : 로그인 여부를 판단하는 인터셉터
  1. JSP
* 화면 View를 구성하는 파일. WEB-INF/views 폴더에 저장하여 사용.

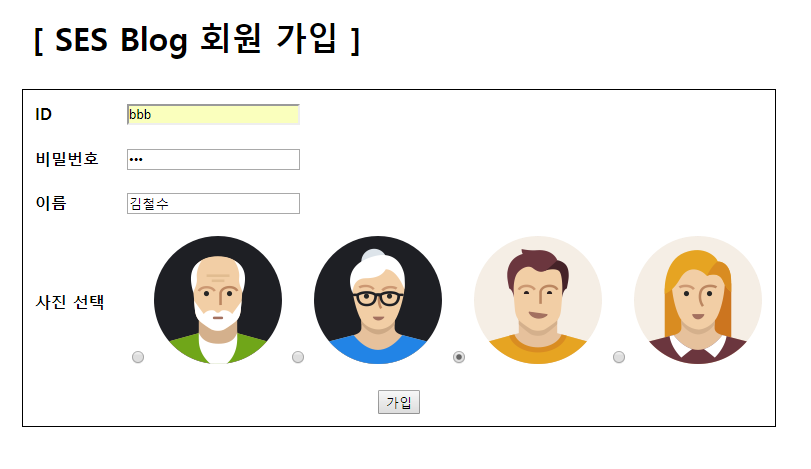
1. 화면 UI
   1. 메인 화면 (로그인 하지 않은 경우)



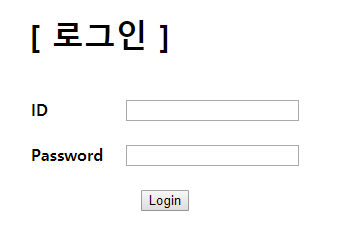
* 1. 메인 화면 (로그인 한 경우)



* 1. 가입 폼



* 1. 로그인



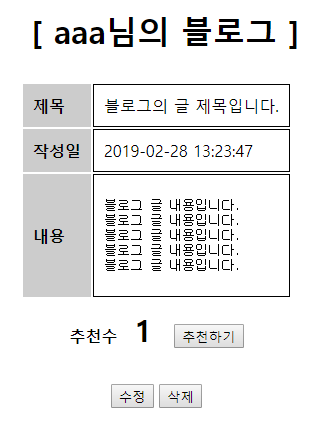
* 1. 모든 회원들의 블로그 목록



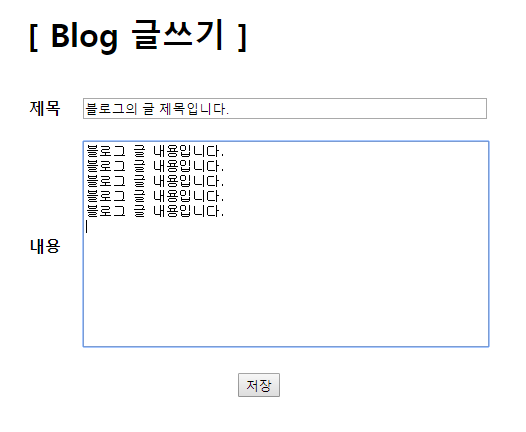
* 1. 특정 회원의 블로그 화면



* 1. 글 읽기



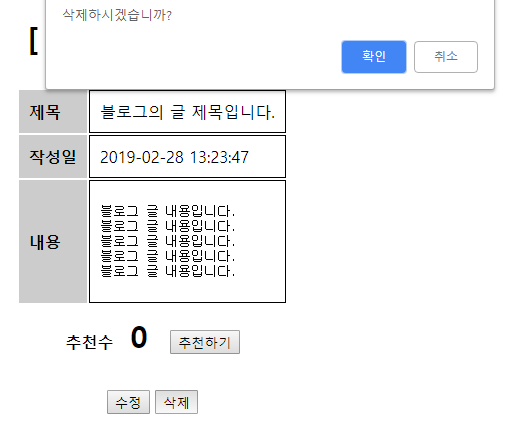
* 1. 글 쓰기



* 1. 글 수정



* 1. 글 삭제



1. 요구사항
   1. 메인 메뉴

* 해당 사이트에 접속하면 “5. 화면 UI”에서 제시한 화면을 출력한다.
* 로그인한 경우와 로그인 하지 않은 경우를 구분하여 메뉴를 표시한다.
* 로그인한 경우에는 화면 상단에 사용자의 이름과 ID를 포함하여 환영 문구를 출력한다.
  1. 회원 가입
* 회원 가입 메뉴를 선택하면 ID와 비밀번호, 이름을 입력하고, 준비된 프로필 이미지 중 하나를 선택할 수 있도록 한다.
* ID는 3~10자로 입력 받는다.
* 비밀번호는 3~10자로 입력 받는다.
* 이름은 반드시 입력하도록 한다.
* 이미지는 1번을 기본 선택 상태로 보여주고, 사용자가 선택한 이미지에 따라 서버로 1,2,3,4 중 하나의 값을 보낸다. (라디오 버튼 이용)
* 위의 조건에 맞지 않게 입력한 경우는 경고창을 표시하고 폼을 Submit하지 않는다.
* 가입 처리가 완료되면 메인화면으로 돌아간다.
  1. 로그인
* ID와 비밀번호를 입력받는다.
* ID나 비밀번호 중의 하나라도 입력하지 않으면 경고창을 표시하고 폼을 Submit하지 않는다.
* ID와 비밀번호가 맞으면 세션에 해당 회원의 ID와 이름을 저장하고 메인화면으로 이동한다.
* ID나 비밀번호가 틀린 경우는 로그인 폼으로 다시 이동한다.
  1. 로그아웃
* 로그아웃 메뉴를 선택하면 세션에 저장된 정보를 모두 삭제하고 메인화면으로 이동한다.
  1. 블로그 목록 보기
* 가입된 모든 회원들의 명단을 ID의 오름차순으로 정렬하여 출력한다.
* 출력할 정보는 프로필 사진, 이름, ID이다.
* 링크를 클릭하면 해당 회원의 글만을 보여주는 개인 블로그 페이지로 이동한다.
  1. 개인 블로그 화면
* 블로그 전체 목록에서 선택한 회원의 ID로 검색하여 그 회원이 작성한 글 목록을 출력한다.
* 로그인한 회원 본인의 블로그인 경우는 화면 하단에 “글쓰기” 버튼을 보여준다.
* 글의 제목을 클릭하면 해당 글의 상세보기 화면으로 이동한다.
  1. 글 상세보기
* 선택한 글의 번호로 검색하여 상세 정보를 보여준다.
* “수정”, “삭제” 버튼은 로그인한 본인 글인 경우에만 표시된다.
  1. 글 수정
* 로그인한 본인이 쓴 글의 “수정” 버튼을 클릭하면 글을 수정할 수 있는 페이지로 이동한다.
* 제목과 내용을 수정하고 다시 “수정” 버튼을 클릭하면 DB에서 글의 정보를 Update하고 다시 글 읽기 화면으로 이동한다.
* 본인 글인 경우에만 수정된다.
  1. 글 삭제
* 로그인한 본인이 쓴 글의 “삭제” 버튼을 클릭하면 대화상자를 띄워 “삭제하시겠습니까?”라는 확인을 받는다.
* 위의 대화상자에서 “확인” 버튼을 누르면 서버로 글 번호를 보내 글을 삭제한다.
* 본인 글인 경우에만 삭제된다.
  1. 추천 기능
* 글의 상세보기 화면에서 “추천하기” 버튼을 누르면 페이지가 이동되지 않고 Ajax를 이용하여 서버로 글의 번호를 보내 해당 글의 추천수를 1 증가시킨다.
* 증가된 추천수를 다시 리턴 받아 화면의 추천수 표시 영역을 갱신한다.
* 추천 기능은 로그인이나 본인글 여부에 관계없이 사용할 수 있다.
  1. Interceptor의 사용
* LoginInterceptor 클래스를 이용하여 글쓰기, 수정, 삭제 기능은 로그인 없이 접근하지 못하도록 처리한다.
* 로그인 하지 않고 위의 경로에 접근했을 때에는 로그인 페이지로 이동시킨다.

1. 채점 기준

|  |  |  |
| --- | --- | --- |
| 구분 | 기능 | 배점 |
| DB | 주어진 조건에 맞게 테이블과 시퀀스를 생성하는 쿼리를 sql파일에 저장 | 5 |
| 메인 화면 | 제시한 화면UI와 같이 로그인 여부에 따라 다르게 출력 | 5 |
| 회원 가입 | 제시한 화면UI와 같이 입력 폼을 출력 | 5 |
|  | 조건에 맞지 않게 정보를 입력한 경우 JavaScript로 경고 대화상자를 보이고 폼을 전송하지 않는다. | 5 |
|  | 사용자가 입력한 정보를 DB에 저장하여 가입 완료 | 5 |
| 로그인 | 제시한 화면UI와 같이 로그인 폼을 출력 | 5 |
|  | 조건에 맞지 않게 정보를 입력한 경우 JavaScript로 경고 대화상자를 보이고 폼을 전송하지 않는다. | 3 |
|  | 로그인 성공한 경우 세션에 정보를 저장하고 메인화면에 사용자 정보 출력 | 5 |
|  | 로그인 실패한 경우 로그인 페이지로 다시 이동 | 5 |
| 로그아웃 | 세션의 모든 정보를 삭제한다. | 5 |
| 블로그 목록 | 가입한 모든 회원의 목록을 출력 | 5 |
|  | 가입시 선택한 프로필 이미지를 사용자 별로 출력 | 3 |
| 개인 블로그 | 해당 회원의 글 목록을 출력 | 5 |
|  | 본인 블로그인 경우에만 “글쓰기” 버튼 표시 | 3 |
| 글 쓰기 | 글을 작성하는 폼을 출력 | 5 |
|  | 입력한 정보와 작성자 아이디를 DB에 저장 | 5 |
| 글 상세보기 | 블로그에서 글 제목을 클릭하면 해당 글의 상세보기 페이지로 이동 | 5 |
|  | 본인 글인 경우에만 “수정”, “삭제” 버튼 표시 | 3 |
| 수정 | 본인 글인 경우에 해당 글을 수정할 수 있는 폼 표시 | 3 |
|  | 수정한 내용을 DB에 Update | 3 |
| 삭제 | 삭제 여부를 묻는 JavaScript 처리 | 3 |
|  | 해당 글을 DB에서 삭제 | 3 |
| 추천 | Ajax를 이용하여 추천수 수정하고 화면 갱신 | 3 |
| Interceptor | 인터셉터를 이용하여 글쓰기, 수정, 삭제 경로에 대한 로그인하지 않는 접근 차단 | 3 |
| 합계 | | 100 |

\* 기본적으로 실행이 안되는 부분은 0점. 관련 기능이 구현되지 않아 실행해 볼 수 없는 부분은 코드가 정상적으로 들어있으면 부분 점수 가능.