데이터베이스및응용 Project #3

Project 제안서

학 과: 컴퓨터공학과

과 목: 데이터베이스및응용

담당교수: 이기훈 교수님

학 번: 2010720019

성 명: 박완배 (1인)

1. **주제 및 설명**

* Social Network Service

웹 어플리케이션 개발주제는 최근 많은 사람들이 사용하고 있는 Social Network Service이다. 그중에서도 가장 인지도가 높은 Facebook을 모델로 삼고 사용될 기능들을 구현하도록 한다. Facebook과 같은 Social Network Service은 사용자가 계속해서 증가하며 컴퓨터, 스마트폰등 많은 디지털 기기를 통해서 언제 어디서나 이용가능하기 때문에 많은 사람들이 모여들고 있다. 그리고 또한 현재 인터넷 시장에서 중요한 위치를 차지하고 있다. Facebook, 트위터, Instagram 등 여러 종류의 SNS의 동작 방식에 대해서 이해하고 이에 대한 Database를 직접 설계하고 구현하도록 한다.

1. **주요 기능**

* 회원가입

|  |
| --- |
|  |

가장 기초가 되면서도 중요한 회원가입 기능이다. 입력으로 사용자의 ID, 이름, 비밀번호 그리고 비밀번호를 확인하며 생년월일, 전화번호 이메일등의 정보를 받는다. 아이디의 경우 Primary Key에 해당하기 때문에 중복된 아이디의 경우 회원가입이 되지 않는다. 또한, 비밀번호 값은 비밀번호 확인 Input 값과 동일한 경우에만 회원 가입이 가능하다. 생년월일이나 전화번호와 같은 값들은 select옵션을 주어서 사용자가 직접 입력하지 않고 편리하게 사용가능하다.

* 로그인

|  |
| --- |
|  |

위의 그림은 메인 페이지로서, 화면 상단에 아이디와 비밀번호 정보를 입력한 후에 로그인 button을 통해서 로그인을 시도하도록 한다. jsp session 객체를 사용했기 때문에 아이디 값은 session이 존재하면 해당 아이디가 자동으로 저장되도록 한다. 이 후, 로그인을 하면 아래의 timeline페이지로 이동하게 된다.

|  |
| --- |
|  |

타임라인 페이지 에서는 main페이지와는 다르게 검색이 가능한 태그가 보여지며, 사용자의 이름이 출력되어서 로그인 한 것을 확인 할 수 있다. 또 한, 이름 옆에는 친구, 로그아웃, 정보수정과 같은 개인정보와 관련된 button들이 나타난다. 그리고 main페이지에서 로그인을 했기 때문에 session객체를 설정하고 session객체를 설정한 경우에는 main page url을 받아도 forwording을 통해서 timeline.jsp파일이 보여지도록 구현을 하였다.

* 타임라인 작성

|  |
| --- |
|  |

다음 기능으로는 타임라인에 게시물을 작성하는 기능이다. 먼저 Textarea에 내용을 입력 할 수 있으며, 하단의 file을 업로드 함으로써 사진을 게시물에 올리고 볼 수 있다. 값을 채워넣은 후에 완료를 누르게 되면, 해당 게시물은 나를 포함한 나의 친구들만 정보를 확인 할 수 있다.

|  |
| --- |
|  |

위의 그림과 같이 업로드한 게시물이 나타나는 것을 확인 할 수 있다. 해당 게시물에는 게시자의 프로필 사진 그리고 이름, 시간 정보를 출력하게 되고 시간 정보에 따라서 가장 최근에 올린 게시물이 제일 위에 오도록 한다.

* 댓 글 기능

|  |
| --- |
|  |

위의 그림은 게시물 하단에 댓 글을 작성하는 기능에 해당한다. 댓 글도 게시물과 같이 최근 작성된 순서로 위에서부터 보이도록 하였다. 또 한, 작성자 본인만 자신의 댓 글을 삭제 할 수 있기 때문에 다른 사용자의 댓 글은 삭제 할 수 없다. 그리고 본인의 이름과 프로필 사진을 통해서 다른 사용자와 원활한 의사소통이 가능하도록 한다.

* 회원 정보 수정

|  |
| --- |
|  |

위의 그림은 회원 정보 수정 화면이다. 기본 회원가입 시 입력 받은 내용들에 대해서 아이디를 제외하고 수정이 가능하다. 비밀번호의 변경 경우에는 현재 비밀번호를 먼저 입력해서 확인을 받고 이 후에 신규 비밀번호와 신규 비밀번호 확인의 값이 일치해야만 변경이 가능하다.

* 친구 목록 확인

|  |
| --- |
|  |

위의 화면은 자신의 친구를 확인하는 화면에 해당한다. 위의 화면을 보면 현재 3명의 사람과 친구 관계이기 때문에 하단 네모에 친구 라고 값이 되어있는 것을 확 인 할 수 있다. 또 한, 상대방의 프로필 사진과 이름을 보고 동명이인인 경우에도 구분 할 수 있기 때문에 매우 유용하다.

* 로그아웃

마지막 로그아웃의 기능은 로그인 시 생성된 session 속성을 제거함으로써 웹 브라우저와 웹 서버간의 session통신을 멈추도록 한다.

1. **사용한 SQL**

* **UserInfoDao.java**

1. select uid from members where uid = ?

위의 Sql문은 Userinfo Dao(Data Access Object)에서 회원가입 시, 이미 존재하는 아이디인지 아닌지를 확인하는데 사용되는 SQL문이다.

1. insert into userinfo(uid,uname,ubirth,ucell,umail) values (?,?,?,?,?,?)”

회원 가입 시 입력된 정보들을 userinfo table에 insert하기 위한 SQL문에 해당한다. 해당 sql문도 UserInfoDao에서 사용되고 있다.

1. select upasswd from userinfo where uid=?

로그인 시 사용자가 기존의 회원이 맞는지 아이디를 확인하고 또 한, 비밀번호도 확인을 하는데 사용이 된다. id값을 통해서 pw값을 불러오기 때문에 확인 여부에 따라서 아이디가 잘못 입력 되었는지 비밀번호가 잘못 입력 되었는지 확인이 가능하다.

1. select \* from userinfo where uid=?

select문을 이용해서 user의 모든 정보를 가져오는 sql문으로 UserInfoDao에 선언되어서 사용되며 한 명의 회원정보를 id를 통해 가져오는데 사용이 된다.

1. update userinfo set uname=?, upasswd=?, ubirth=?, ucell=?, umail=? where uid=?

회원 정보를 수정하는 경우에 사용되는 method에 포함된 sql문이다. uid값을 통해서 나머지 회원의 정보를 업데이트 시킨다.

* **BoardDao.java**

1. select \* from board order by b\_time DESC

타임라인의 게시물들을 오름차순 시간순서로 불러오는데 사용이 된다. 하나 이상의 row가 불러와 질 수 있기 때문에 자바 라이브러리 ArrayList를 이용해서 BoardDto ArrayList를 생성해서 반환하는 메소드에서 사용이 되었다.

1. insert into board(b\_id, b\_text, b\_img, b\_time) values(?,?,?,?)

게시물을 작성해서 데이터베이스에 저장하는데 사용되는 SQL문이다. board 테이블의 Not null값들을 채운다.

* **CommentDao.java**

1. select \* from comment where b\_no=? order by ctime DESC

게시물에 대한 댓 글들을 불러오는 메소드에서 사용되는 SQL문이다. 하나의 게시물에 여러 개의 댓 글이 생성 될 수 있기 때문에 ArrayList로 선언해서 CommentDto클래스 변수들을 반환하도록 한다.

1. insert into comment(b\_no, ctext,cid,ctime) values (?,?,?,?)

댓 글 입력 시 사용되는 SQL문이다.

1. delete from comment where cno=?

댓 글 번호 primary key인 cno값을 통해서 댓 글을 삭제하는 SQL문이다.

* **FDao.java**

1. select U.uid, U.uname, U.upasswd, U.ubirth, U.umail, U.uimg from (friend as F JOIN userinfo as U ON my\_id=?) where F.fr\_id=U.uid

나의 친구목록을 찾는데 사용되는 SQL문이다. Inner JOIN을 사용하지 않는 경우에는 2번의 SQL 질의를 해야 하는 반면에 Inner JOIN을 통해서 friend 테이블과 userinfo 테이블 중에서도 나의 아이디와 일치하면서 친구의 아이디가 userinfo table에 있는 회원들을 불러오는데 사용된다. 한가지 주의사항으로는 친구관계를 insert할 때 양방향으로 설정해 주어야 한다.

1. **YouTube링크**

<https://www.youtube.com/watch?v=ATzOH8K1FWQ&t=1s>

1. **고찰**

이번 웹 어플리케이션을 구현하면서 완성도가 많이 낮았던 것은 사실이다. 설계했던 것 보다 많이 구현을 못하였고 SQL문도 다양하게 사용하지 못한 아쉬움이 남는다. 하지만 한가지 확실히 배배운 것은 이번 웹 어플리케이션을 구현하면서 jsp & Servlet을 사용해서 구현한 경험이라는 점이다. 사실 2차 프로젝트에서 개발환경에 대해서 Spring을 이용해서 구현을 하겠다 목표를 세웠지만, 인터넷과 서적들을 통해서 java, jsp & Servlet만을 소화하는데도 굉장히 많은 시간이 소요되었다. 그리고 jsp & Servlet에서도 MVC 패턴을 이용한 Frontcontroller, Command 방식을 많이 활용하지 못한 점이 아쉬움이 남지만 앞으로도 계속해서 개발을 하면서 소화해내는 것이 목표이다.

그리고 이번에 jsp & Servlet을 접하면서 처음 DAO와 DTO에 대해서 알게 되었는데 이와 같이 Database 모델에 따라서 설정을 해두고 개발을 하는 것이 굉장히 편하다는 것을 알게 되었고 직접 구현해 보았기 때문에 매우 좋은 경험이 되었다. 또 한, java 환경이기 때문에 여러 가지 libaray들을 직접 찾아보고 응용한 점에서 스스로 만족을 느끼고 있다. bootstrap의 경우에는 본인이 원하는 대로 디자인을 만들 수 없는 것이 굉장히 아쉬웠지만 작은 것 하나하나까지 신경을 쓰다 보니 큰 구조에 대해서 놓친 것 같다. 이번 계기를 통해서 디자인도 중요하지만, 우선 기능을 먼저 어느 정도 구현을 한 후에 집중적으로 디자인을 한다면 개발 시간도 절약 할 수 있을 것으로 생각된다. 부족함이 더 많은 프로젝트였지만 많은 것을 배울 수 있었던 것 같다.

1인 프로젝트로 진행을 하였습니다. 여럿이서 개발을 하다 보면 혼자 도맡아 하던 기억이 많아서 한번 처음 설계부터 구현까지 해보고자 하는 욕심이 있었으나 이루지 못해 아쉬웠습니다.