

<인터넷 프로그래밍 기말과제 설계보고서>

12171588 김민석

1. 목표 : 게임 - 총알피하기

총알에 맞지 않고 오랫동안 버티는 것이 목적인 게임을 구현하고자 한다.

구현해야 하는 객체	효과 및 특성(요구사항)
플레이어 캐릭터(비행기)	방향키로 이동, 총알과 닿으면 게임종료
총알	직선으로 이동, 벽에 반사, 주기적으로 추가됨
별	게임화면을 구성하는 별 객체

구현할 기능들
회원가입(아이디 중복 방지)
로그인
비밀번호 변경(개인정보 수정)
게임 플레이
Database 연동(Mybatis 이용)
순위변동 및 관리
관리자 기능(user 관리)
명예의 전당(1 위 표시)

2. 화면 설계 및 흐름

<화면 목록>

1. 오프닝 화면 - index.jsp, start.jsp	2. 회원가입 화면 - register.jsp, registerFailed.jsp, registerSuccess.jsp
3. 로그인 화면 - login.jsp, loginIdFailed.jsp, loginPwFailed.jsp, loginPwModified.jsp	4. 도움말 화면 - help.jsp
5. 관리자 로그인 화면 - adminlogin.jsp, adminFailed.jsp	6. 일반 User 페이지 - mypage.jsp, mypageFailed.jsp
7. 관리자 페이지 - adminpage.jsp	8. 게임시작 - gameplay.jsp

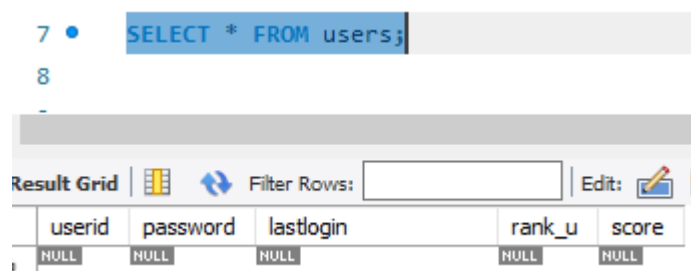
추가로 객체들을 전달하는 페이지인 authentication.jsp 페이지가 있는데, 이는 화면 상에

노출되지 않는다. 또한, 페이지 처리 및 데이터베이스와의 연동 등을 담당하는

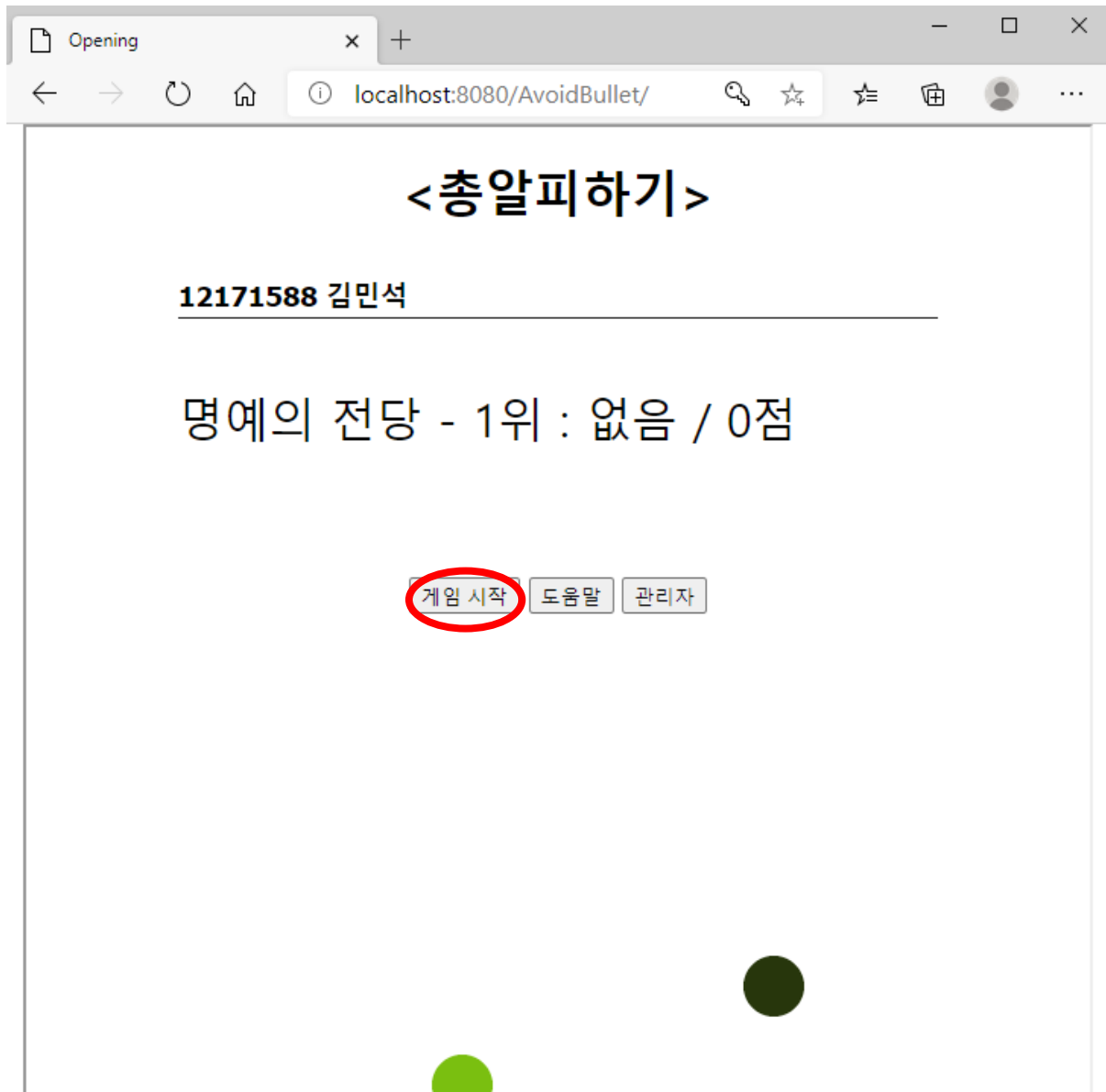
UserController.java, UserDao.java, User.java, DBUtil.java 파일이 있다.

<화면 흐름>

- 초기 DB 상태



- 오프닝 화면



접속 시에 가장 먼저 보이는 화면이다. index.jsp 에서 iframe 을 하나 생성한 후에, 이 곳에 start.jsp 를 띄우는 방식이다.

```
<iframe src="authentication.jsp?action=start" width="700" height="700" scrolling="no">
</iframe>
```

위의 스크립트로 인해 authentication.jsp 로 페이지가 이동된다. 그리고, get 방식으로 보낸 파라미터를 받아서 authentication.jsp 가 상황에 따라 객체 파라미터들을 처리한다. 그리고 주어진 상황에 맞는 페이지로 forwarding 해주기 위해 UserController.java 에 파라미터 값을 넘겨주면서 자바코드를 실행한다.

```
<jsp:forward page="UserController?userid=${ID}&password=${PW}&state=${STATE}${ADDITIONAL}"></jsp:forward>
```

위 코드를 통해 UserController.java 에게 흐름을 넘긴다. 이후 UserController.java 에서는 id, pw, state 등을 전달받고, state 값에 따라 각각의 페이지로 이동하게 된다.

이 방식으로 전체적인 페이지의 요청을 처리하게 된다. 단, 페이지 이동 시에 값의 전달이 필요 없는 상황이면 단순히 위치를 지정해 주는 방식으로 구현했다.

어떤 페이지에서 버튼이나 이벤트 등으로 인해 값을 전달하면서 페이지를 이동해야 할 때에는 먼저 authentication.jsp 에 get 및 post 방식으로 값을

전달한다. 그리고 이어서 상황에 맞게 파라미터를 조절하여 UserController.java 로 값을 전송한다. 마지막으로 자바 코드에서 state 변수 등을 판단하여 적절한 페이지에 올바른 객체들을 전달하고 forward 시켜줌으로써 해당 페이지를 보여준다.

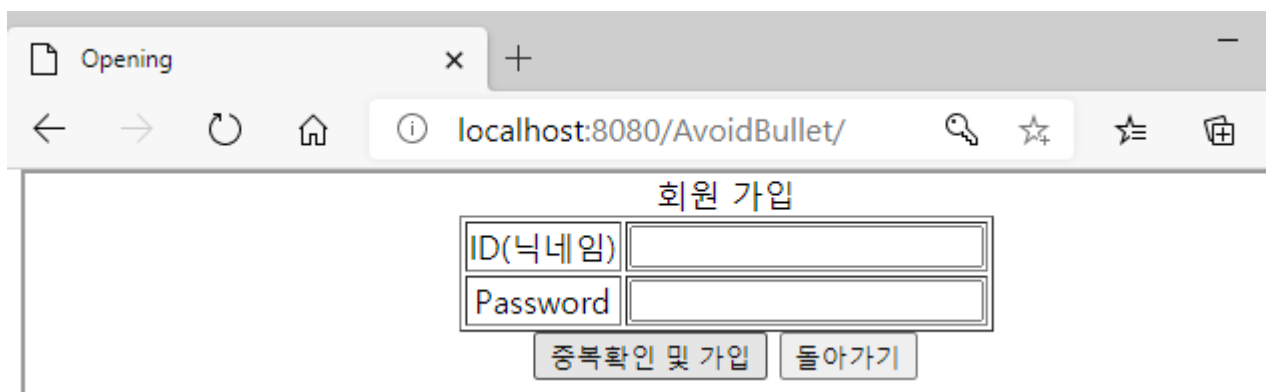
또한, UserController.java 에서 데이터베이스를 다룰 때에는 UserDao.java 를 이용한다. 이는 자바에서 데이터베이스 쿼리를 효율적으로 수행하기 위해 UserMapper.xml 이라는 파일과 연동하여서 쿼리실행을 도와주는 클래스이다.

- 로그인 화면

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/AvoidBullet/'. The page content is a login form titled 'Login'. It contains two input fields: 'ID(닉네임)' and 'Password'. Below these fields are three buttons: '로그인', '회원가입' (which is circled in red), and '돌아가기'.

게임시작 버튼을 클릭하면 login.jsp 로 이동한다 여기서 ID 와 Password 를 입력 받고, 로그인 시에 해당 값을 전달하면서 로그인을 처리하게 된다.

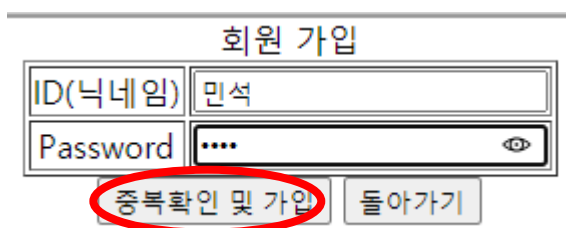
- 회원가입 화면



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/AvoidBullet/'. The page title is '회원 가입' (Member Registration). The form contains two input fields: 'ID(닉네임)' (ID/Nickname) and 'Password'. Below the fields are two buttons: '중복확인 및 가입' (Check for duplicates and register) and '돌아가기' (Go back).

회원가입 버튼을 클릭하면 해당 페이지인 register.jsp 로 이동한다.

- 회원 가입 진행



The screenshot shows the same '회원 가입' (Member Registration) form. The 'ID(닉네임)' field contains the text '민석'. The 'Password' field contains four dots. The '중복확인 및 가입' (Check for duplicates and register) button is circled in red.

Login

ID(닉네임)	<input style="width: 95%;" type="text"/>
Password	<input style="width: 95%;" type="password"/>
<input type="button" value="로그인"/> <input type="button" value="회원가입"/>	

회원가입 성공!

회원 정보를 입력하고, '중복확인 및 가입' 버튼을 누르면 회원가입 로직이 진행된다. 이 때에는 authentication.jsp 에 id, password 값과 회원가입 상태임을 알리는 'register' state 를 파라미터로 전송하고, 그대로 UserController.java 가 받아서 ID 중복 확인, DB 에 입력, 로그인 성공/실패 페이지로 forward 등의 절차를 거쳐 회원가입을 진행한다.

- 일반 User 페이지

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/AvoidBullet/'. The page content is as follows:

My Page

플레이어 : 민석
 현재 순위 : 1
 최고 점수: 0
 최근 접속: "2020-12-19 06:22:38"

<비밀번호 변경란>

현재 비밀번호 입력 :
 새로운 비밀번호 입력 :

login 페이지에서 회원정보를 올바르게 입력하면 mypage.jsp 로 이동한다.

로그인에 실패한 경우, loginIdFailed.jsp/loginPwFailed.jsp 등을 띄우며

로그인에 실패했음을 알린다. My Page 에서는 현재 접속한 user 의 이름,

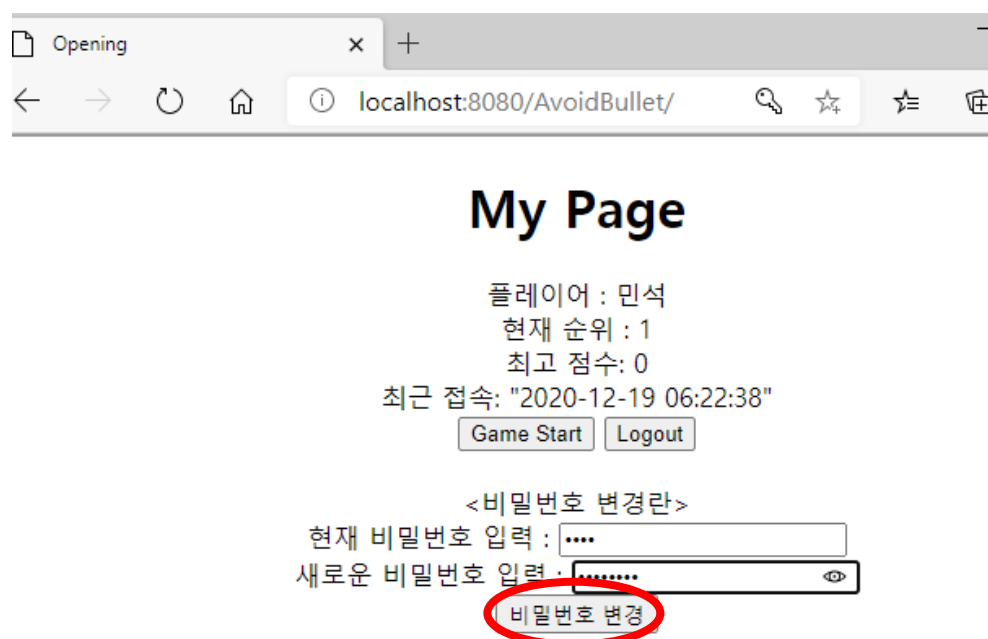
순위, 최고점수, 최근접속 등의 정보를 보여주고, 게임을 시작하거나

비밀번호 변경, 로그아웃 등을 수행할 수 있다. login.jsp 에서는 id, password,

state 값 및 user 객체를 전송하여 해당 사용자의 정보를 mypage 에서

읽을 수 있다.

- 비밀번호 변경



Opening x +

localhost:8080/AvoidBullet/

My Page

플레이어 : 민석
현재 순위 : 1
최고 점수 : 0
최근 접속: "2020-12-19 06:22:38"

Game Start Logout

<비밀번호 변경란>

현재 비밀번호 입력 :

새로운 비밀번호 입력 :

비밀번호 변경

Opening x +

localhost:8080/AvoidBullet/

Login

ID(닉네임)

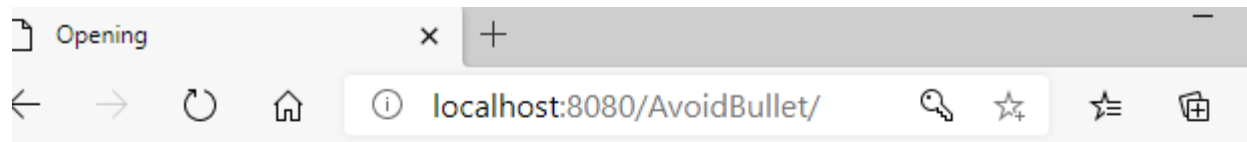
Password

Password 변경 성공! 다시 로그인 해주세요.

만약 위 그림과 같이 현재 비밀번호가 올바르게 입력되었다면, Password 변경 성공을 알리는 로그인 페이지인 loginPwModified.jsp 로 이동한다.

만약 비밀번호 변경 란에서 누락된 요소가 있거나, 현재 비밀번호가 잘못 입력되었을 경우에는 비밀번호 변경이 진행되지 않는다. 이와 같이 데이터베이스를 다루는 부분에서의 로직은 대부분 UserController.java 에서 처리하도록 구현했다.

- 게임 시작



플레이어 : 민석

현재 순위 : 1

현재 점수: **1111**

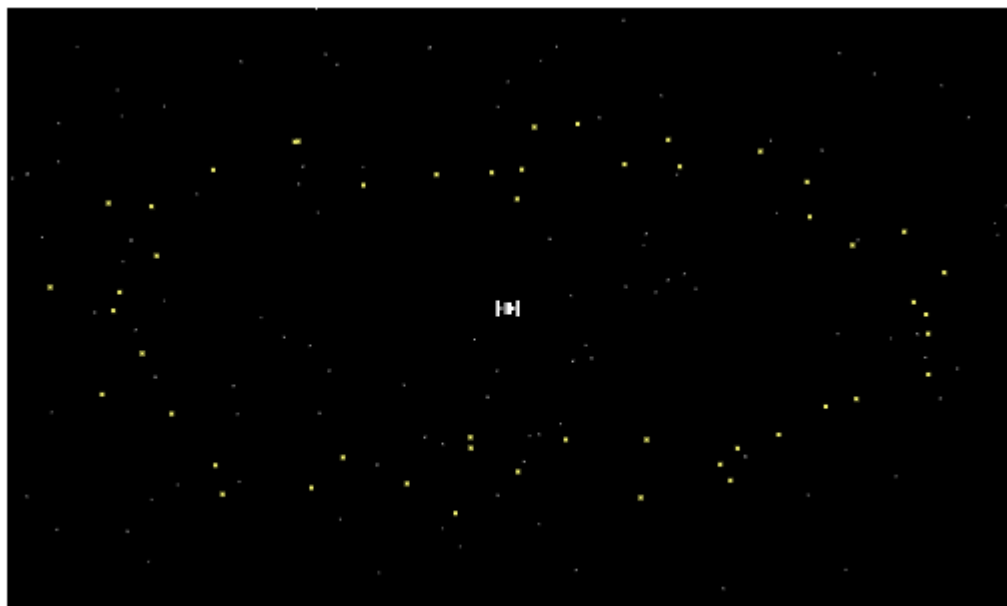
Last Login: "2020-12-19 06:41:01"

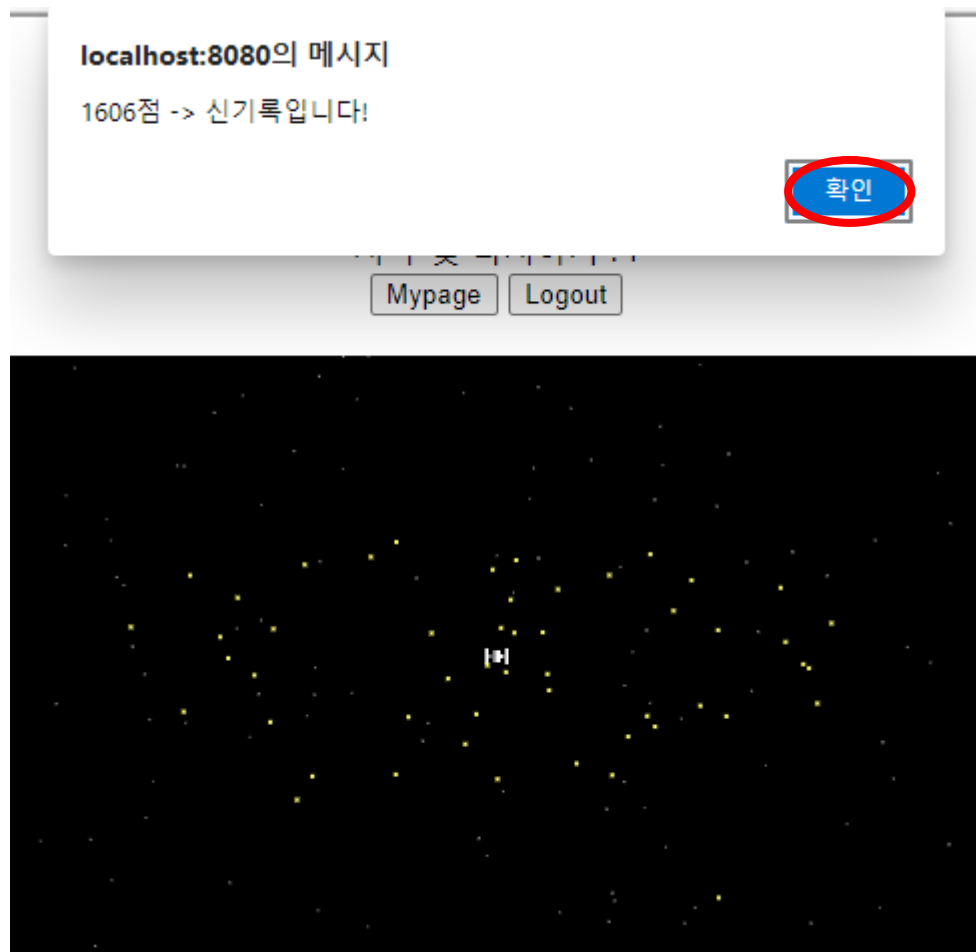
Rank: "1"

시작 및 다시하기 : r

[Mypage](#)

[Logout](#)





My Page

플레이어 : 민석

현재 순위 : 1

최고 점수: 1606

최근 접속: "2020-12-19 06:41:01"

Game Start

Logout

<비밀번호 변경란>

현재 비밀번호 입력 :

새로운 비밀번호 입력 :

비밀번호 변경

My Page 에서 Game Start 버튼을 클릭하면 게임 시작 화면으로 이동한다.

바로 게임이 시작되며, r 키를 통해 재시작을 할 수 있다. 사용자의 정보를

보여주고, 점수가 점차적으로 올라간다. 이 곳에서 전체적인 게임 한 판이 진행된다고 볼 수 있으며, 비행기나 총알, 배경, 별 등은 모두 canvas 를 이용하여 구현했다.

비행기는 방향키를 통해 움직일 수 있으며, 총알에 맞으면 게임이 종료된다. 게임이 종료되면 두 가지의 이동 경로가 있다. 점수가 신기록을 달성했을 경우에는, 알림 창을 띄우고 나서 MyPage 로 이동한다. 이 때 보여지는 순위 및 점수는 이전 게임 결과가 적용된 이후의 정보이다. 만약 점수가 최고기록보다 낮을 경우에는 그 상태로 화면이 멈추며, r 키나 Mypage, Logout 버튼 등을 통해 다른 이벤트를 실행할 수 있다.

- 로그아웃

<총알피하기>

12171588 김민석

명예의 전당 - 1위 : 민석 / 1606점

게임 시작

도움말

관리자



로그아웃을 통해 처음화면으로 돌아오면, 처음에 생성했던 이름이 명예의 전당에 등록되어 있는 것을 볼 수 있다. 이제 새로운 이름으로 회원가입을 진행해서 1 위의 기록을 깨는 과정을 진행해볼 것이다.

- 새로운 사용자(user1) 회원가입 : 표시한 부분을 클릭하면서 새로운 회원가입 과정을 진행한다.

<총알피하기>

12171588 김민석

명예의 전당 - 1위 : 민석 / 1606점

게임 시작

도움말

관리자

Login

ID(닉네임)

Password

로그인

회원가입

돌아가기

회원 가입

ID(닉네임)

user1

Password

중복확인 및 가입

돌아가기

Login

ID(닉네임)

Password


로그인

회원가입

회원가입 성공!

돌아가기

Login

ID(닉네임)	user1
Password 
<input type="button" value="로그인"/> <input type="button" value="회원가입"/>	

회원가입 성공!

My Page

플레이어 : user1

현재 순위 : 2

최고 점수: 0

최근 접속: "2020-12-19 06:52:54"

<비밀번호 변경란>

현재 비밀번호 입력 :

새로운 비밀번호 입력 :

위의 과정을 진행하면, user1 이 새로 생성되었고, 순위가 현재 2 위임을 알 수 있다. 이후 게임 진행을 통해 기존에 존재하는 1 위의 기록을 깨는 과정은 다음과 같다.

플레이어 : user1

현재 순위 : 2

현재 점수: **792**

Last Login: "2020-12-19 06:54:30"

Rank: "2"

시작 및 다시하기 : r

[Mypage](#)

[Logout](#)



localhost:8080의 메시지

5863점 -> 신기록입니다!

확인

Mypage

Logout



My Page

플레이어 : user1

현재 순위 : 1

최고 점수: 5863

최근 접속: "2020-12-19 06:56:32"

Game Start

Logout

<비밀번호 변경란>

현재 비밀번호 입력 :




새로운 비밀번호 입력 :

비밀번호 변경

Mypage 로 돌아온 후, 순위와 점수가 변동되어 있는 것을 확인할 수 있다.

이 때 mysql 에서 users 테이블을 조회해보면 다음과 같다.

```
7 • SELECT * FROM users;
8
```

result Grid   Filter Rows: <input type="text"/> Edit: 				
userid	password	lastlogin	rank_u	score
user1	asdf	2020-12-19 06:57:28	1	5863
민석	asdfasdf	2020-12-19 06:47:30	2	1606
NULL	NULL	NULL	NULL	NULL

이후 Mypage 에서 Logout 을 클릭하여 처음화면으로 돌아가면, 다음과

같이 명예의 전당에 1 위가 바뀌어 있음을 확인할 수 있다.

<총알피하기>

12171588 김민석

명예의 전당 - 1위 : user1 / 5863점

게임 시작

도움말

관리자



- 관리자 페이지에서 사용자 관리 진행 : 처음 화면에서 '관리자'버튼을 클릭하면 관리자 로그인 페이지로 이동한다.

Admin Login

ID

PW

로그인

돌아가기

Admin Login

ID

kms

PW

.....

로그인

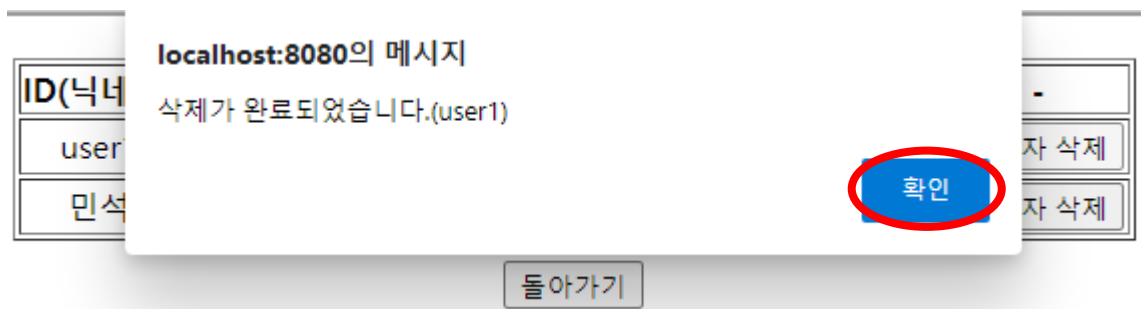
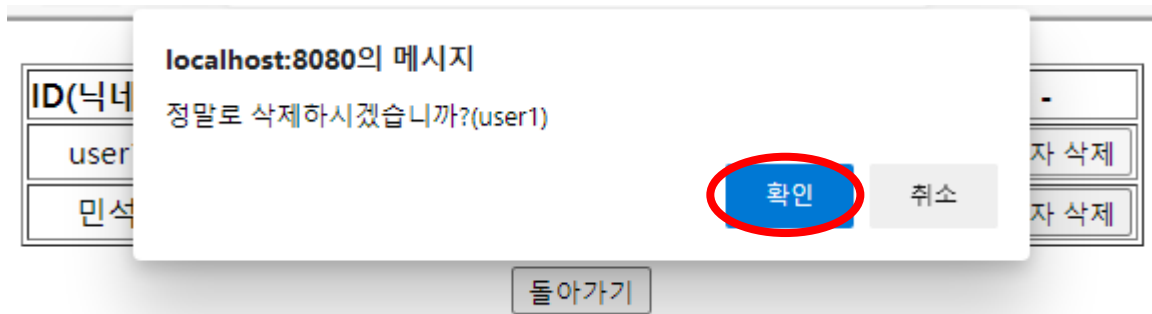
돌아가기

관리자의 ID 와 PW 는 각각 kms, 12171588 이다. 이를 입력하고 로그인 버튼을 누르면 관리자페이지로 이동한다. 만약, 누락된 항목이 있거나 잘못된 정보를 입력했을 경우에는 관리자페이지로 이동이 되지 않는다.

- 사용자 삭제 진행 : 아래 그림은 관리자페이지의 모습이다. 모든 사용자가 점수의 순위별로 정렬된 형태를 표에 나타낸 모습이다. '사용자 삭제' 버튼을 누르면, 정말 삭제할 것인지 물어보는 확인창이 나타나고, 확인을 누르면 해당 가로줄의 user 를 삭제한다.

<Users>					
ID(닉네임)	Password	최근 로그인 시간	전체 순위	최고점수	-
user1	asdf	2020-12-19 06:57:28	1	5863	사용자 삭제
민석	asdfasdf	2020-12-19 06:47:30	2	1606	사용자 삭제

돌아가기

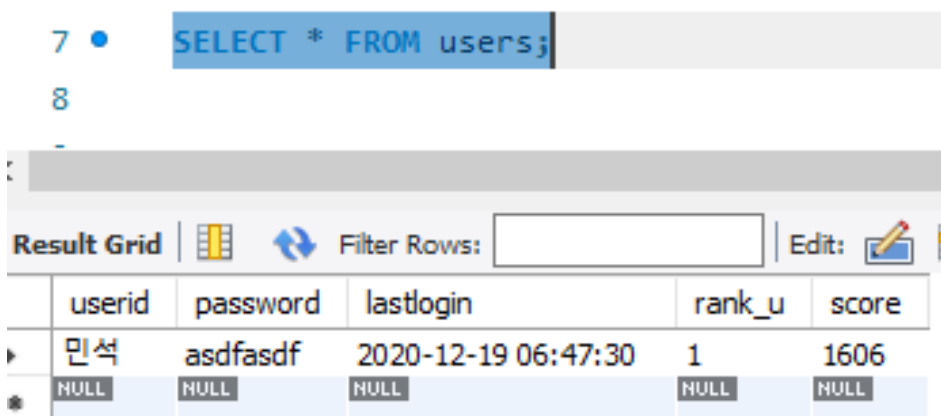


<Users>

ID(닉네임)	Password	최근 로그인 시간	전체 순위	최고점수	-
민석	asdfasdf	2020-12-19 06:47:30	1	1606	사용자 삭제

[돌아가기](#)

이 때의 DB 를 mysql Workbench 로 확인해보면 다음과 같다.



이후, 남은 사용자도 삭제하면 users 테이블에 사용자정보가 모두 사라지게 된다.

<Users>

ID(닉네임)	Password	최근 로그인 시간	전체 순위	최고점수	-
민석	asdfasdf	2020-12-19 06:47:30	1	1606	사용자 삭제

돌아가기

localhost:8080의 메시지

정말로 삭제하시겠습니까?(민석)

확인

취소

localhost:8080의 메시지

삭제가 완료되었습니다.(민석)

확인

<Users>

ID(닉네임)	Password	최근 로그인 시간	전체 순위	최고점수	-
---------	----------	-----------	-------	------	---

현재 등록된 사용자가 없습니다.

돌아가기

<총알피하기>

12171588 김민석

명예의 전당 - 1위 : 없음 / 0점

게임 시작

도움말

관리자





이렇게 해서 가장 처음에 접속했던 상태로 다시 돌아오게 되었다. 이 때의

DB 테이블은 다음과 같다.

```
7 • SELECT * FROM users;
```

```
8
```

Result Grid				
 Filter Rows: <input type="text"/>				
Edit: 				
userid	password	lastlogin	rank_u	score
NULL	NULL	NULL	NULL	NULL

3. 데이터베이스 설계

<users>

Mybatis 를 사용하여 구현했으며, 연동한 DB 의 이름은 userdb 이다. 테이블에 대해 설명하면, 사용자 이름인 userid, 비밀번호 password, 최근 로그인 시간 lastlogin, 순위 rank_u, 최고점수 score 를 속성으로 가지는 users 라는 table 을 생성한다.

테이블 생성 쿼리는 다음과 같다.

```
CREATE TABLE `userdb`.`users` (  
  
  `userid` varchar(100) NOT NULL,  
  `password` varchar(100) NOT NULL,  
  `lastlogin` TIMESTAMP DEFAULT NOW(),  
  `rank_u` int unsigned DEFAULT NULL,  
  `score` int unsigned DEFAULT NULL,  
  PRIMARY KEY (`userid`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8
```


JAVA 와 DB 를 연동할 때 사용된 파일은 mybatis-config.xml, DBUtil.java 파일이다. mybatis-config.xml 에서 mysql 의 username, password 를 각각 root, mysql 로 지정해주었고, 쿼리를 담은 xml 파일을 UserMapper.xml 로 설정해주었다. DBUtil.java 는 mybatis-config.xml 파일 경로를 지정하고, 이를 통해 SQL Session 을 연결할 때 이용되는 클래스이다. 이후 자바 코드에서 DB 관련 쿼리를 실행할 때에는 UserDao.java 클래스에서 UserMapper.xml 에 정의된 쿼리의 id 로 접근하여 실행하게 된다.

4. 개발 일정

날짜	프론트 엔드	백 엔드
11 주차 (11.9~11.13)	화면 레이아웃 설계	각 이벤트 처리
12 주차 (11.16~11.20)		디비 연동 및 구축
13 주차 (11.23~11.27)		디비 보완, 서버 연결
14 주차 (11.30~12.4)	디자인	
15 주차 (12.7~12.11)	로그인 구현	
16 주차 (12.14~12.18)	부족한 부분 보완	