<총알피하기>

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(1위 표시) |

1. 화면 설계 및 흐름

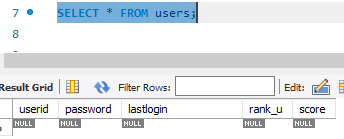
<화면 목록>

|  |  |
| --- | --- |
| 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 User페이지  - 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.jsp에서 iframe을 하나 생성한 후에, 이 곳에 start.jsp를 띄우는 방식이다.



위의 스크립트로 인해 authentication.jsp로 페이지가 이동된다. 그리고, get 방식으로 보낸 파라미터를 받아서 authentication.jsp가 상황에 따라 각 객체 파라미터들을 처리한다. 그리고 주어진 상황에 맞는 페이지로 forwarding해주기 위해 UserController.java에 파라미터 값을 넘겨주면서 자바코드를 실행한다.



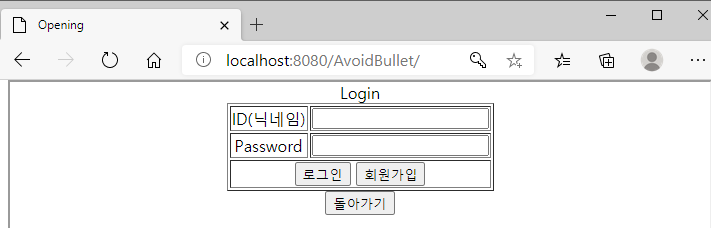
위 코드를 통해 UserController.java에게 흐름을 넘긴다. 이후 UserController.java에서는 id, pw, state 등을 전달받고, state값에 따라 각각의 페이지로 이동하게 된다.

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.jsp에 get 및 post 방식으로 값을 전달한다. 그리고 이어서 상황에 맞게 파라미터를 조절하여 UserController.java로 값을 전송한다. 마지막으로 자바 코드에서 state 변수 등을 판단하여 적절한 페이지에 올바른 객체들을 전달하고 forward시켜줌으로써 해당 페이지를 보여준다.

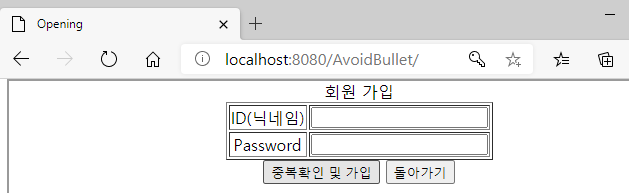
또한, UserController.java에서 데이터베이스를 다룰 때에는 UserDao.java를 이용한다. 이는 자바에서 데이터베이스 쿼리를 효율적으로 수행하기 위해 UserMapper.xml이라는 파일과 연동하여서 쿼리실행을 도와주는 클래스이다.

- 로그인 화면



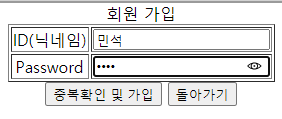
게임시작 버튼을 클릭하면 login.jsp로 이동한다 여기서 ID와 Password를 입력 받고, 로그인 시에 해당 값을 전달하면서 로그인을 처리하게 된다.

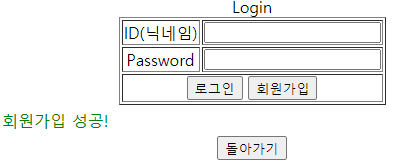
- 회원가입 화면



회원가입 버튼을 클릭하면 해당 페이지인 register.jsp로 이동한다.

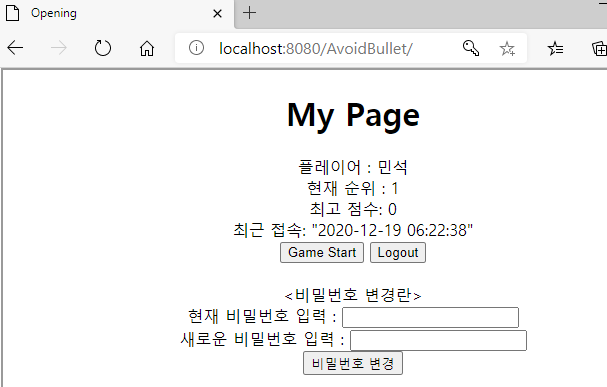
- 회원 가입 진행





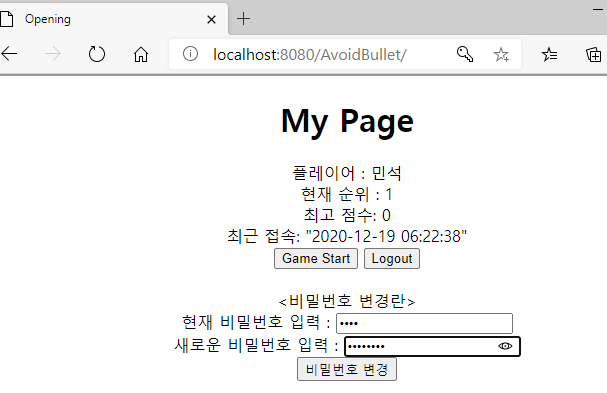
회원 정보를 입력하고, ‘중복확인 및 가입’ 버튼을 누르면 회원가입 로직이 진행된다. 이 때에는 authentication.jsp에 id, password값과 회원가입 상태임을 알리는 ‘register’ state를 파라미터로 전송하고, 그대로 UserController.java가 받아서 ID중복 확인, DB에 입력, 로그인 성공/실패 페이지로 forward 등의 절차를 거쳐 회원가입을 진행한다.

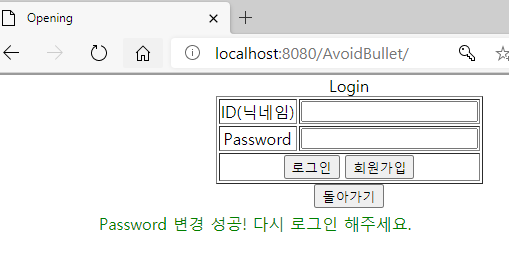
- 일반 User페이지



login페이지에서 회원정보를 올바르게 입력하면 mypage.jsp로 이동한다. 로그인에 실패한 경우, loginIdFailed.jsp/loginPwFailed.jsp 등을 띄우며 로그인에 실패했음을 알린다. My Page에서는 현재 접속한 user의 이름, 순위, 최고점수, 최근접속 등의 정보를 보여주고, 게임을 시작하거나 비밀번호 변경, 로그아웃 등을 수행할 수 있다. login.jsp에서는 id, password, state 값 및 user 객체를 전송하여 해당 사용자의 정보를 mypage에서 읽을 수 있다.

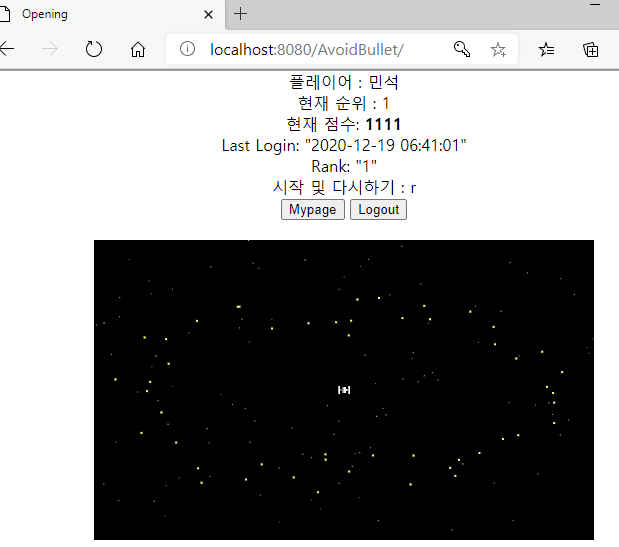
- 비밀번호 변경

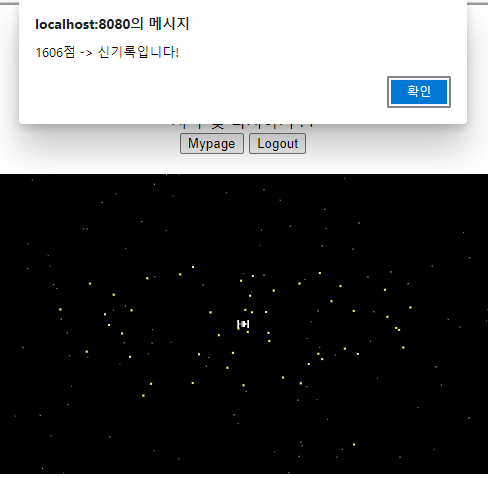




만약 위 그림과 같이 현재 비밀번호가 올바르게 입력되었다면, Password 변경 성공을 알리는 로그인 페이지인 loginPwModified.jsp로 이동한다. 만약 비밀번호 변경 란에서 누락된 요소가 있거나, 현재 비밀번호가 잘못 입력되었을 경우에는 비밀번호 변경이 진행되지 않는다. 이와 같이 데이터베이스를 다루는 부분에서의 로직은 대부분 UserController.java에서 처리하도록 구현했다.

- 게임 시작







My Page에서 Game Start버튼을 클릭하면 게임 시작 화면으로 이동한다. 바로 게임이 시작되며, r키를 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 canvas를 이용하여 구현했다.

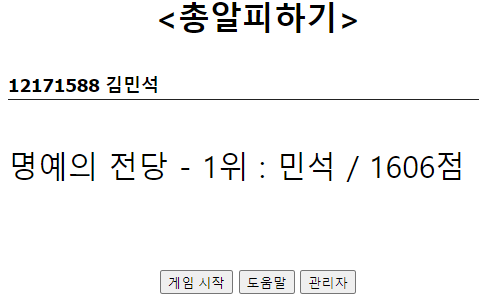
비행기는 방향키를 통해 움직일 수 있으며, 총알에 맞으면 게임이 종료된다. 게임이 종료되면 두 가지의 이동 경로가 있다. 점수가 신기록을 달성했을 경우에는, 알림 창을 띄우고 나서 MyPage로 이동한다. 이 때 보여지는 순위 및 점수는 이전 게임 결과가 적용된 이후의 정보이다. 만약 점수가 최고기록보다 낮을 경우에는 그 상태로 화면이 멈추며, r키나 Mypage, Logout버튼 등을 통해 다른 이벤트를 실행할 수 있다.

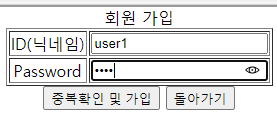
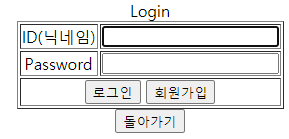
- 로그아웃

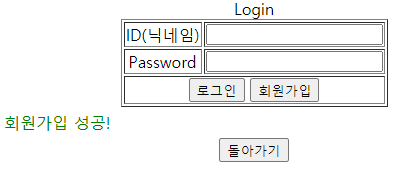


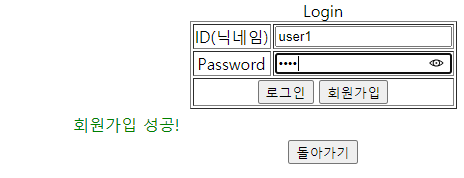
로그아웃을 통해 처음화면으로 돌아오면, 처음에 생성했던 이름이 명예의 전당에 등록되어 있는 것을 볼 수 있다. 이제 새로운 이름으로 회원 가입을 진행해서 1위의 기록을 깨는 과정을 진행해볼 것이다.

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



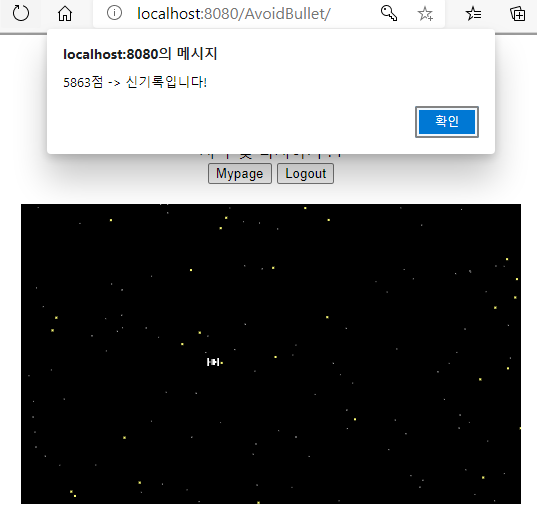






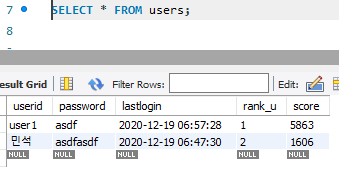


위의 과정을 진행하면, user1이 새로 생성되었고, 순위가 현재 2위임을 알 수 있다. 이후 게임 진행을 통해 기존에 존재하는 1위의 기록을 깨는 과정은 다음과 같다.

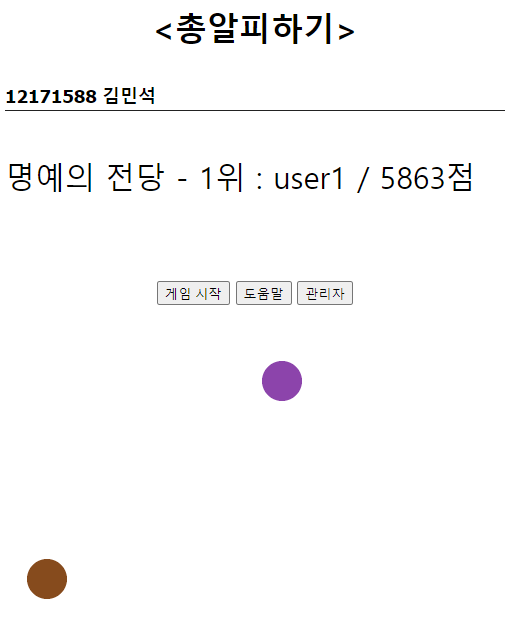




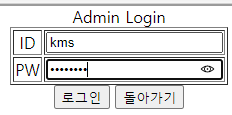
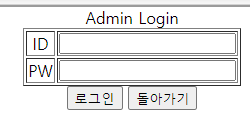
Mypage로 돌아온 후, 순위와 점수가 변동되어 있는 것을 확인할 수 있다. 이 때 mysql에서 users 테이블을 조회해보면 다음과 같다.



이후 Mypage에서 Logout을 클릭하여 처음화면으로 돌아가면, 다음과 같이 명예의 전당에 1위가 바뀌어 있음을 확인할 수 있다.

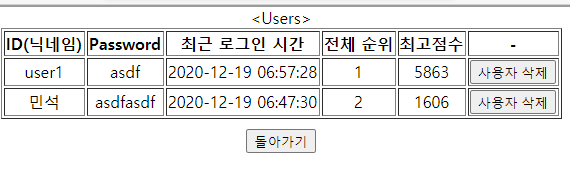


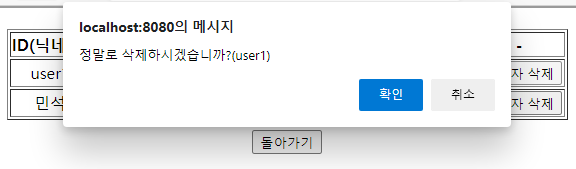
- 관리자 페이지에서 사용자 관리 진행 : 처음 화면에서 ‘관리자’버튼을 클릭하면 관리자 로그인 페이지로 이동한다.

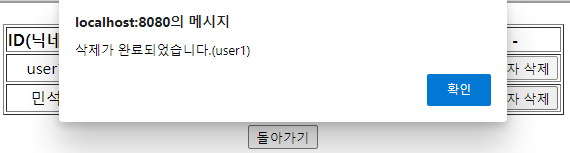


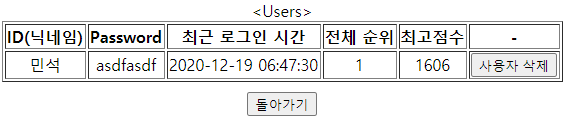
관리자의 ID와 PW는 각각 kms, 12171588이다. 이를 입력하고 로그인 버튼을 누르면 관리자페이지로 이동한다. 만약, 누락된 항목이 있거나 잘못된 정보를 입력했을 경우에는 관리자페이지로 이동이 되지 않는다.

- 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 user를 삭제한다.

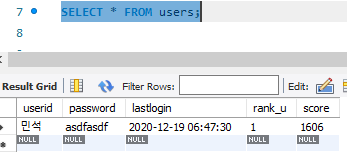




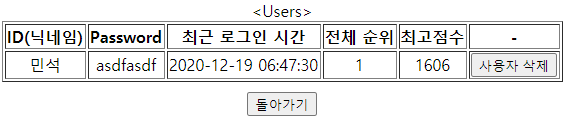


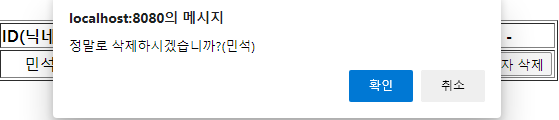


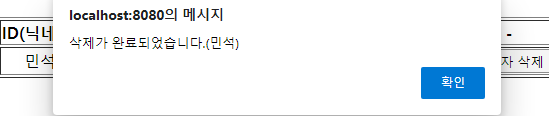
이 때의 DB를 mysql Workbench로 확인해보면 다음과 같다.

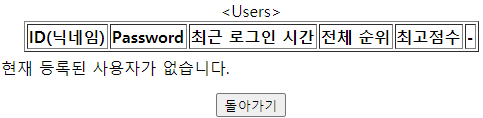


이후, 남은 사용자도 삭제하면 users 테이블에 사용자정보가 모두 사라지게 된다.



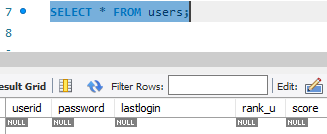








이렇게 해서 가장 처음에 접속했던 상태로 다시 돌아오게 되었다. 이 때의 DB테이블은 다음과 같다.

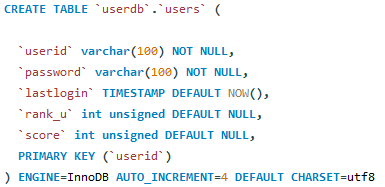


1. 데이터베이스 설계

<users>

Mybatis를 사용하여 구현했으며, 연동한 DB의 이름은 userdb이다. 테이블에 대해 설명하면, 사용자 이름인 userid, 비밀번호 password, 최근 로그인 시간 lastlogin, 순위 rank\_u, 최고점수 score를 속성으로 가지는 users라는 table을 생성한다.

테이블 생성 쿼리는 다음과 같다.



JAVA와 DB를 연동할 때 사용된 파일은 mybatis-config.xml, DBUtil.java파일이다. mybatis-config.xml 에서 mysql의 username, password를 각각 root, mysql로 지정해주었고, 쿼리를 담을 xml파일을 UserMapper.xml로 설정해주었다. DBUtil.java는 mybatis-confit.xml파일 경로를 지정하고, 이를 통해 SQL Session을 연결할 때 이용되는 클래스이다. 이후 자바 코드에서 DB관련 쿼리를 실행할 때에는 UserDao.java클래스에서 UserMapper.xml에 정의된 쿼리의 id로 접근하여 실행하게 된다.

1. 개발 일정

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