

인터넷 프로그래밍 시간에 배운 내용들을 바탕으로 Tomcat 서버에서 Firebase를 기반으로 하는 웹 프로젝트를 제작하였다. Eclipse IDE로 코드를 작성하였고, Tomcat 서버는 8.0버전을 사용하였다. 인하대 후문 주변의 맛집을 회원이 선택한 기준에 따라 추천해주는 사이트를 만들었다. 그 날 먹을 음식점을 정하기 어려울 때 누군가 대신 추천해주길 원했었는데, 이번 기말 프로젝트를 수행하면서 직접 만들어보았다. 11개의 jsp와 2개의 servlet을 만들어 구현하였다.

Tomcat 서버를 시작하고 가장 먼저 뜨는 창은 1.jsp가 되도록 하였다. 인터페이스를 기준으로 보면 제목과 설명 밑에 “뭐 먹지?”가 있고, ‘클릭하여 로그인’을 클릭하면 로그인창이 뜨게 하였다. 그 옆에는 캔버스로 만든 시계가 있고, 맨 하단에는 관리자 버튼이 있다.

관리자 버튼을 누르고 관리자 비밀번호 12161595를 입력하면 admin.jsp 관리자 창으로 이동한다. 관리자 창에서는 등록된 회원들의 모든 정보와 가게들의 모든 정보를 전부 확인 할 수 있다. 특정 회원이나 가게의 정보를 수정하거나 삭제할 수 있으며, 회원 등록과 가게 등록도 가능하다. 수정과 삭제를 클릭했을 때 id 또는 sid를 함수에 전달해 그 아이디에 해당되는 정보만 조정할 수 있도록 하였다. 가게를 등록하는 survey.jsp에서는 양식에 맞춰 가게를 등록하고, 그 아래에서 바로 확인할 수 있도록 가게들의 정보를 다시 한 번 띄웠다. 회원가입과 비슷한 방식으로 user_survey에 입력한 정보를 sid에 따라 set 하면 등록이 완료된다. 가게 정보의 수정은 sProfile.jsp에서 이루어지는데, 가게의 정보를 먼저 불러오고, update로 수정된 정보로 갱신될 수 있게 하였다. 삭제는 confirm으로 수행여부를 다시 한 번 확인하고, 특정 child 데이터를 remove로 삭제하도록 하였다. 수정되거나 삭제됨에 따라 firebase에서는 실시간으로 값이 변화되므로 관리자 창에서도 실시간으로 값의 변화를 확인할 수 있다.

1.jsp에서 ‘클릭하여 로그인’을 클릭하면 frm함수가 실행되고, frm함수는 로그인이 되어있지 않는, 즉 id가 널 값을 가지는 상태일 때는 iframe의 src가 로그인 창인 welcome.jsp가 되고, 로그인이 되어있으면 LoginPage.jsp로 연결되도록 설정하였다.

welcome.jsp에서는 아이디와 비밀번호를 입력하여 로그인하거나 회원가입을 할 수 있는 JoinUs.jsp로 이동할 수 있는데, 회원가입 후 로그인을 할 수 있다.

JoinUs.jsp는 폼의 양식에 따라 회원의 정보를 입력하고 완료 버튼을 누르면 가입이 되도록 구현하였는데, 이때 아이디는 유일해야 하므로 mycheck()로 중복여부를 확인하고 가입할 수 있도록 설정하였다. user_data속 push된 회원의 id와 pw, user_profile의 동일 id의 정보를 입력된 값들로 set하면 회원가입이 완료된다.

로그인시 firebase의 DB속 user_data의 childsnapshot에서 아이디와 비밀번호 값이 입력된 값과 일치하면 로그인이 성공하고, LoginServlet으로 아이디와 비밀번호 값을 전송하도록 하였다. LoginServlet에서는 session을 가져와서 parameter를 attribute로 설정하고, LoginPage.jsp로 전송하도록 하였다. 로그인이 성공하면, id를 받아와 user_data와 user_profile에서 일치하는 id의 정보를 불러오고, innerHTML로 html 내부에 입력되도록 하였다. 회원의 이름과 전공, 학년, 성별, 그리고 등록한 음식점을 불러오고, 로그아웃 하거나 개인정보 수정, 탈퇴를 할 수 있다. 로그아웃은 logout.jsp에서 session의 attribute를 제거하고 유효하지 않게 설정한 뒤 다시 로그인 창으로 돌아간다. 개인정보 수정은 Profile.jsp에서 이루어지는데, user_profile속 회원의 기존 정보를 불러와서 수정할 수 있도록 하였다. update로 값이 변경 되고나면 수정이 완료된다.

다시 1.jsp의 iframe내에서 로그인이 성공하여 LoginServlet으로 정보가 넘어갔음이 확인되

면, 즉 id가 널 값을 가지지 않으면, “뭐 먹지?”를 눌렀을 때 1-1.jsp로 이동한다. 로그인 창 옆은 캔버스로 시계를 구현하였다. 현재 시간을 받아와서 매 초, 분, 시마다 중심을 기준으로 원 모양으로 회전할 수 있도록 하였다.

1-1.jsp는 session의 회원 이름을 가져와 띄워주고, 품의 양식에 맞춰 그 날의 먹고 싶은 음식과 평소 취향에 따라 설문지를 작성하는 페이지이다. 제목 밑에는 유튜브 소스를 가져와 음악을 들으며 설문지를 선택할 수 있도록 하였다. 설문지는 다양하지만 구체적인 가게를 찾을 수 있도록 구현하였고, 시간은 10분부터 60분 사이의 선택한 값 이상의 가게를, 가격은 1000원부터 10000원 사이의 선택한 값 이하의 가게를 추천하도록 하였다. 제출 버튼을 누르면 search()에서 입력된 값을 user_survey의 데이터베이스에서 검색하고, 정해진 기준에 따라 일치하는 값을 찾았을 시 그 값의 아이디, 즉 가게 아이디인 sid와 음식점 이름을 SurveyServlet에 전송한다. SurveyServlet에서는 LoginServlet과 동일하지만 user_survey의 정보를 attribute로 설정할 수 있도록 구현하였다.

1-2.jsp에서는 1-1에서 일치했던 음식점의 이름을 띄우고, resources 폴더에 음식점의 이름으로 저장되어있는 이미지 파일을 불러온다. 그리고 DB에 있던 음식점의 정보 중 대표메뉴와 가격, 식사시간을 불러와서 소개를 덧붙였다. ‘네이버 지도’를 클릭하면 하단의 iframe의 src가 네이버 지도의 링크를 불러오는데, 이때 ‘용현동 +가게이름’으로 검색할 수 있도록 설정하여 보다 정확한 위치를 찾을 수 있도록 구현하였다. 그리고 ‘검색 링크’를 클릭하면 새 창으로 용현동의 가게이름이 검색된 네이버 창이 뜨도록 하였다. 이 가게를 등록하고 싶으면 등록 버튼을 누르면 되고, 누르면 user_data속 회원의 user_last가 이 가게 이름으로 update 된다. 회원이 이 가게가 마음에 들면 등록할 수 있게 함으로써 다시 로그인했을 때 자신이 등록한 가게를 확인할 수 있도록 하였다. 등록 후 1.jsp로 돌아가지고, ‘클릭하여 로그인’을 눌렀을 때 본인의 로그인 정보 속 ‘가장 최근 등록한 가게’가 갱신되었음을 확인할 수 있다.

사이트를 처음부터 구상하고 구현하는 과정에 시행착오가 많았으나, 이 프로젝트를 통해 jsp와 서블릿이 어떻게 유기적으로 연결되어 값을 전송하고 전달받는지 알 수 있었고, firebase에서 insert, update, delete를 수행하는 방법에 대해 배울 수 있었다. 그리고 회원과 관리자의 입장에서 어떻게 하면 인터페이스가 편리하고 한 눈에 들어오는 디자인으로 만들어질지 고민해볼 수 있었다. 실제로 내가 사용할 유용한 사이트를 만들었다는 것에 큰 성취감을 얻을 수 있었다.