**Final Report**

201321310 심현준

201221318 김기상

1. **Web Smu talk의 개요**

프로그램을 다운 받아서 쓰는 메신저와 다르게 누구나 쉽게 접근 할 수 있는 웹에서 최소한의 정보를 입력해서 가입을 하여 사용합니다. 입력하였던 학과를 매칭해서 같은 학과의 사람들을 친구 등록 할 수 있도록 제공해 주고, 친구를 등록한 리스트 중에 원하는 대상을 더블 클릭하게 되면 메시지를 보내고 받을 수 있는 room이 제공됩니다. 그 외에 자신이 메시지를 보내고 받았던 room들의 리스트를 보여주는 창과 그룹채팅을 진행 할 수 있게 선택할 수 있는 창, 즐겨찾기를 등록할 수 있는 창이 제공됩니다.

1. **Web Smu talk의 기능**

제안서에서는 오프라인과 온라인을 구분하기로 했었지만, 조언에 의한, DB를 사용해서 메시지 내용을 저장하는 방법으로 구분하지 않고 진행 하였습니다.

**[ 로그인 ]**

아이디와 비밀번호를 입력 받고, 아이디와 비밀번호가 비어있지 않는지 검사를 진행합니다. 비어있지 않으면 아이디와 비밀번호를 검사해서 맞는지 확인합니다.

**[ 가입 ]**

이름, 학과, 아이디, 비밀번호 이렇게 4가지의 최소한의 정보만을 입력 받습니다. 입력 후 가입 버튼을 누르게 되면 해당아이디가 존재하는지 검사를 진행 하게 됩니다.

**[ 프로필 리스트 ]**

기본 프로필 사진과 이름이 있는 테이블들의 리스트를 가집니다. 자신과 자신이 등록한 친구들을 볼 수 있습니다.

**[ 메신저 room list ]**

자신과 친구의 1:1 혹은 group으로 메시지를 주고 받았던 room들의 리스트들을 보여주고, 더블 클릭을 하게 되면 해당 room이 나타나게 됩니다.

**[ 친구 등록 ]**

자신과 학과가 같은 사람들의 리스트를 제공해 줌으로써 친구를 등록 할 수 있도록 해줍니다.

**[ 즐겨 찾기 ]**

자신과 친구를 등록한 사람들 중에서 자주 대화하는 사람들을 선택하여 프로필 리스트에서 즐겨 찾기 되지 않는 친구들의 리스트보다 상단에 위치하도록 합니다.

**[ 메신저(1:1 / group) ]**

프로필 리스트에서 원하는 대상을 더블 클릭하여 메시지를 주고 받을 수 있는 room을 만듭니다. 만들어진 room은 대상의 오프라인과 온라인에 상관없이 메시지를 보낼 수 있으며, 대상이 온라인일 경우 채팅의 모습처럼 대화가 진행됩니다.

그룹으로 메시지를 보낼 대상들을 선택하여 room을 만들게 되면 다수의 사람에게 메시지를 전송 할 수 있으며, 온라인인 대상과는 채팅의 모습처럼 진행됩니다.

1. **Web Smu talk 설계**

**[DB 설계]**

-미리 만들어진 table-

|  |  |  |  |
| --- | --- | --- | --- |
| membership\_information (회원 정보) | | | |
| id (아이디) | password (비밀번호) | name (이름) | department (학과) |

|  |  |  |
| --- | --- | --- |
| friend (자신이 추가한 친구정보) | | |
| id (아이디) | FriendId (친구 아이디) | favorites (즐겨찾기) |

|  |  |  |  |
| --- | --- | --- | --- |
| chat (메시지 room들에 대한 정보) | | | |
| source  (송신 아이디) | chatoom  (메시지 room 이름) | participation  (개설 날짜와 시간) | destination  (수신 아이디) |

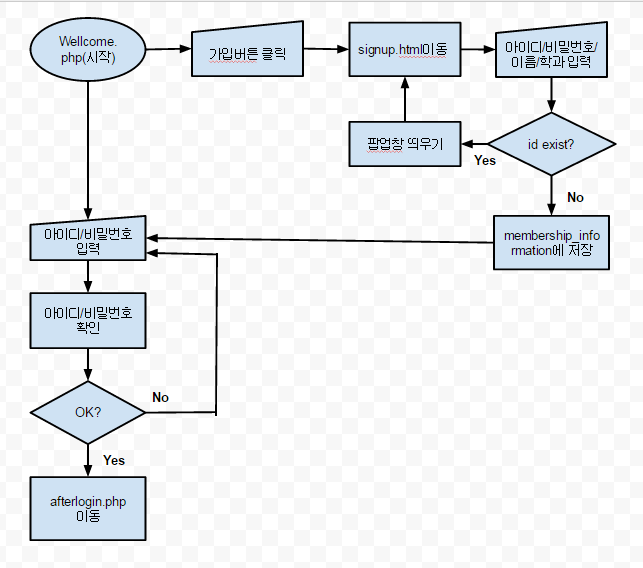
-동적으로 만들어지는 table-

여기서 chatroom이라는 이름은 개설한 시간 + 개설한 사람의 아이디가 합쳐져서 동적으로 만들어 집니다. 하지만 지금은 임의로 chatroom이라고 하겠습니다.

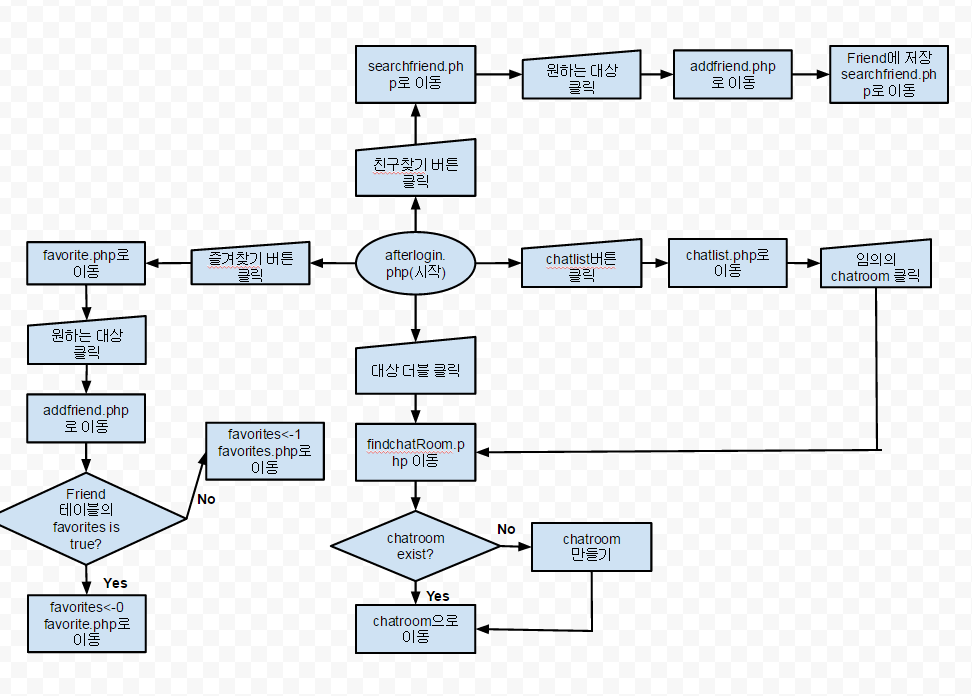
|  |  |  |  |
| --- | --- | --- | --- |
| chatroom (메시지 room) | | | |
| Id  (송신 아이디) | name  (이름) | message  (메시지 내용) | time  (보낸 시간) |

**[ flow chart ]**

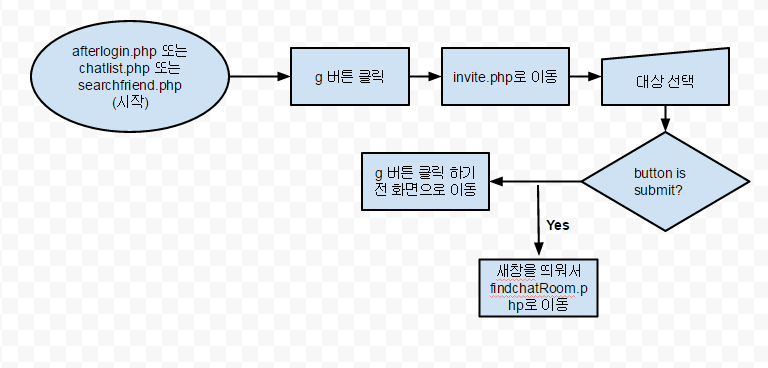
-로그인 / 가입 part-

****

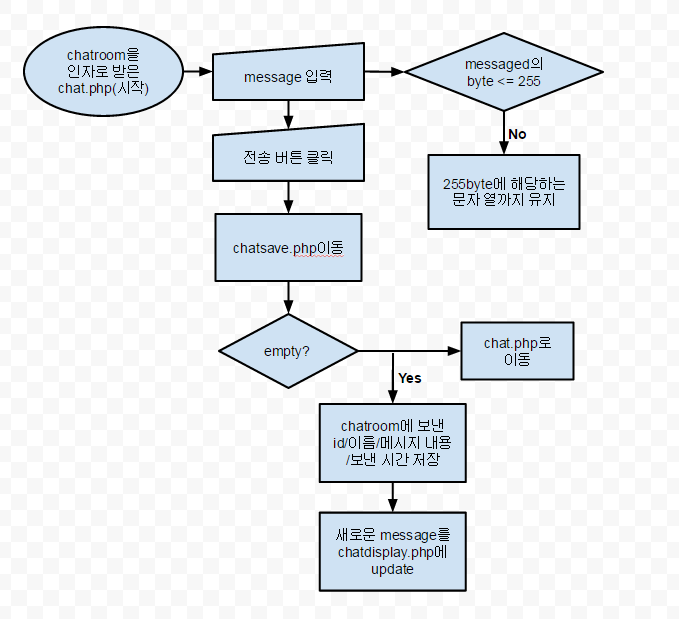
-로그인 후 첫 화면인 afterlogin.php에서 시작-



-group 버튼 클릭 후-



-chat.php에서 시작-



1. **Web Smu talk 구현**

Mysql, html, css, javascript, php를 사용하여 위에서 설계한 것과 생각한 기능들을 토대로 만들었습니다.

**[ Mysql ]**

create table membership\_information(

id char(12) not null,

password char(16),

name char(40),

department char(25),

primary key(id))default charset=utf8;

create table friend(

id char(12) not null,

FriendId char(12),

favorites tinyint(1))default charset=utf8;

alter table friend add foreign key friend(id) references membership\_information(id);

create table chat(

source char(12) not null,

chatroom char(22),

participation char(10),

destination char(12))default charset=utf8;

alter table chat add foreign key chat(source) references membership\_information(id);

한글을 저장 할 수 있도록 위와 같이 utf8로 table을 만들기 위해서 my.ini을 수정 하였습니다.

-my.ini에서 수정된 부분-

[mysqldump]

quick

max\_allowed\_packet = 16M

character-set-server=utf8

[mysql]

no-auto-rehash

default-character-set=utf8

[mysqld]

port=3306

explicit\_defaults\_for\_timestamp = TRUE

character-set-client-handshake=FALSE

init\_connect="SET collation\_connection = utf8\_general\_ci"

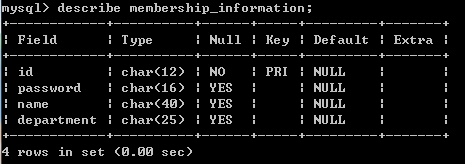
init\_connect="SET NAMES utf8"

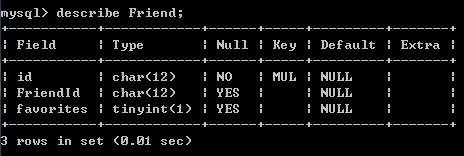
character-set-server=utf8

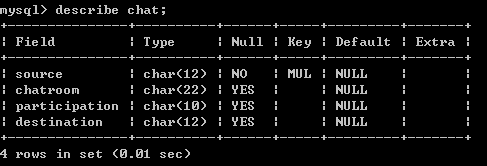
[client]

default-character-set = utf8

**[DB 구현된 모습]**







**[Code 부분]**

각 페이지의 주요 코드들을 중심으로 진행하겠습니다.

먼저 가장 첫 번째 화면인 Wellcome.php부터 시작하겠습니다.

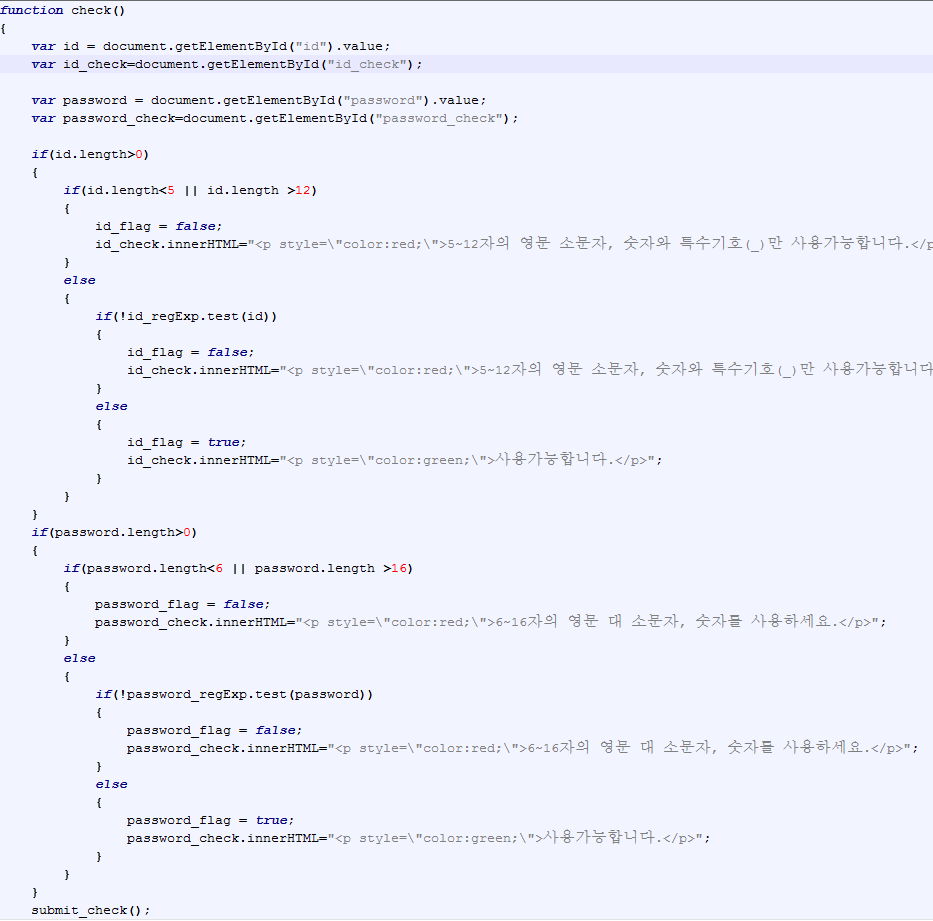
-Wellcome.php-



만약에 로그인을 하게 되면, $\_COOKIE[‘cookie’]값에 login이라는 값이 들어가게 됩니다. 그 후에 Wellcome.php를 다시 요청하게 되면 $\_COOKIE[‘cookie’]값이 비어있지 않기 때문에 afterlogin.php로 바로 연결되고, 로그인을 하지 않은 상태거나 혹은 로그인 이후에 브라우저를 완전히 종료하고 Wellcome.php를 요청하게 될 경우 해당 페이지를 보여줍니다. 이 페이지에서는 로그인을 할 수 있도록 아이디와 패스워드를 입력 할 수 있는 공간이 있고, 로그인 버튼, 가입 버튼이 있습니다.

-signup.html-





가입 화면에서는 간단하게 이름, 학과, 아이디, 비밀번호를 입력을 받게 구성이 되어있습니다. 아이디 또는 비밀번호를 입력한 후 정해진 공간에서 마우스로 클릭하여 check()함수를 활성 되게 하면, 폼에 입력한 아이디 혹은 비밀번호의 id값을 받아와서 길이를 비교하고 조건을 만족하면 정규식에 있는 범위에 벗어나는 값들인지 확인합니다. 만약에 벗어난다면 inner\_HTML을 활용해서 아이디, 혹은 비밀번호 밑에 <p>태그를 넣어주어서 만족 시켜야 하는 조건이 무엇인지 알려주게 됩니다. 그리고 밑에 submit\_check()은 모든 조건이 만족 하면 전송버튼을 활성화 시킨다는 함수입니다.

-makeId.php-



전송 버튼을 누른 후에 해당 페이지로 이동하게 되는데, 이 페이지에서는 id와 password값이 비어있는지 확인합니다. 문제 없다면 membership\_information 테이블에 접근해서 입력한 id와 같은 값을 가지는 id가 존재하는지 확인하고 있으면 signup.html로 이동시킴과 동시에 id가 이미 존재한다는 내용의 팝업 창을 띄워줍니다. 존재하지 않으면 membership\_information에 insert문을 통해서 입력한 폼들의 값을 저장해 줍니다.

-loginConfirm.php-



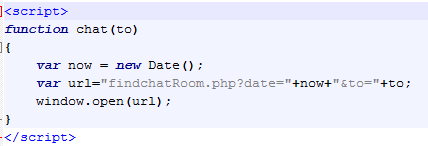
아이디와 비밀번호를 입력한 후에 로그인 버튼을 클릭하게 되면 loginConfirm.php가 호출되는데, 먼저 아이디와 비밀번호가 비어있는 값으로 들어왔는지 확인해줍니다. 문제가 없을 경우 membership\_information으로 접근해서 해당 아이디가 존재하는지 확인합니다. 존재할 경우, select문을 통해서 받아온 비밀번호와 입력한 비밀번호를 비교합니다. 모든 조건을 만족 하게 되면 $\_SEESION에 자신이 들어온 ip주소를 인덱스로 하여 id값을 저장하게 됩니다. 그리고 쿠키의 값도 저장해주고 브라우저를 완전히 종료하기 전에Wellcome.php를 요청하게 되면 로그인 이후의 화면인 afterlogin.php가 나오도록 합니다.

성공적으로 로그인이 되었다는 의미로 header()을 통해서 afterlogin.php로 이동하게 됩니다.

-afterlogin.php-







로그인을 하게 되면 처음으로 보이는 페이지입니다. 먼저 기본 프로필 사진, 자신의 이름을 포함하는 프로필 영역을 만들기 위해서 membership\_information에 session에 저장했던 자신의 아이디와 같은 아이디를 찾고, 해당 아이디와 관련된 정보들을 받아옵니다. 그리고 <table>을 이용해서 프로필 영역을 만듭니다.

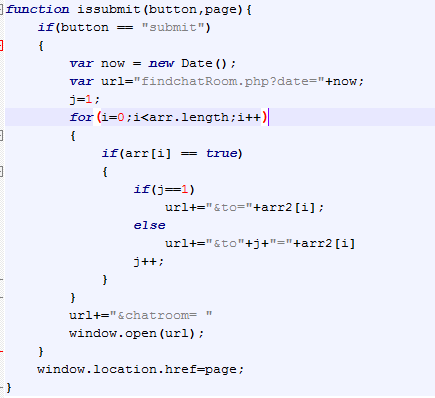
자신이 즐겨찾기로 등록한 친구들이 있다면, 즉 Friend테이블의 favorites이 true(1)이라면 자신의 프로필 밑이고, 친구의 리스트들의 위에 존재할 수 있도록, 먼저 만들어주게 됩니다. Friend테이블에 접근해서 자신의 아이디로 친구를 추가한 사람들의 아이디를 받아오게 됩니다. 그리고 친구들의 수 만큼 membership\_information에 접근해서 친구들의 정보를 받아오고 <table>을 이용해서 프로필 영역을 만들게 됩니다. 이번에는 자신의 프로필 영역을 만드는 것과 달리 chat()이라는 javascript함수가 해당 영역을 더블 클릭하면 호출 될 수 있도록 합니다.

그리고 즐겨찾기와 상관없이 자신의 아이디로 친구 추가한 사람들의 id를 모두 받아와서 위와 같은 과정을 똑같이 진행하게 됩니다.

chat()함수는 더블 클릭하게 되면 이 함수를 호출한 시간과 인자로 base64로 encode 된 친구의 아이디를 받아와서 findchatRoom.php에 넘겨주게 됩니다.

-invite.php-





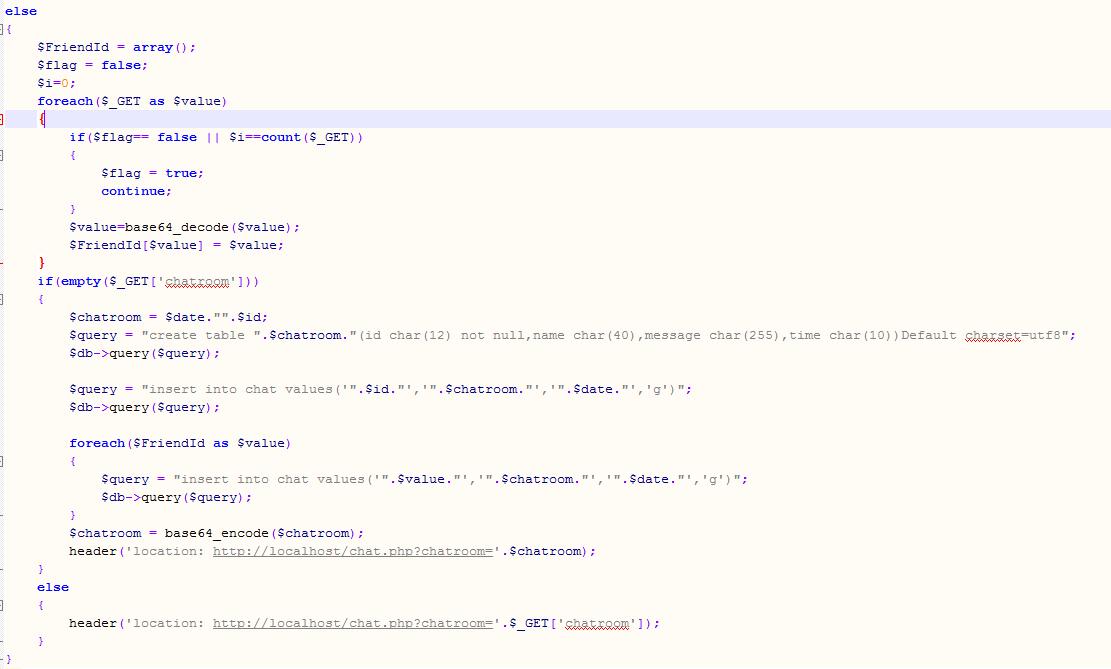
makeflag() 함수는 그룹 채팅의 대상을 보여주기 위해서 친구들의 리스트를 만들 때 친 수가 넘어가게 되는데, 그 수만큼 flag를 저장할 수 있는 배열을 구성하고 false값을 저장 하게 됩니다. 이와 비슷하게 friend()함수도 base64로 encode 된 친구의 아이디를 인자로 받아서 배열에 저장하게 됩니다. 이 두 함수를 통해서 값을 저장하고 만들어진 배열들은 대상이 선택 되었는가 선택되지 않았는가를 판단하게 되는 지표가 됩니다.

isclick()함수는 친구의 프로필 영역을 클릭하게 되었을 때, 클릭되었다고 알려줘야 하기 때문에 해당 프로필 영역의 배경색과 클릭된 이미지로 바꿔주게 되고, flag값을 true로 바 꿔줍니다.

그리고 확인버튼을 눌렀을 경우 issubmit함수의 button인자에 submit이 들어가게 됩니다. 만약에 취소버튼을 눌렀을 경우에는 cancel값이 들어가서 해당 if문을 실행하지 않고 window.location.href=page; 구문만을 실행합니다. 이 구문은 page인자, 즉 g버튼을 눌러서 invite.php를 호출시킨 페이지의 url이 들어있어서 그 페이지로 이동할 수 있도록 합니다. 확인버튼이면 if문을 실행해서 새 창으로 열게 될 url을 구성하게 됩니다. url은 findchatRoom.php의 이름과 확인버튼을 눌렀을 때의 시간, 선택한 대상들의 encode 된 아이디들을 포함하게 됩니다.

-findchatRoom.php-





findchatRoom.php에는 2개 이상의 인자들이 넘어오게 되는데, 먼저 javascript의 new Data()를 통해서 얻은 시간은 필요 없는 문자열도 포함하고 있기 때문에 explode(“GMT”,$date)하여서 앞부분을 가지고 strtotime을 통해서 년 월 일 시 분 초의 문자열을 10개의 길이를 가진 숫자들로 구성된 문자열로 만들어 줍니다. 인자가 4개 이하일 경우에는 1대1 이므로, 자신의 아이디와 친구의 아이디가 참여하고 있는 chatroom이 있는지 chat테이블에 질의를 하고 있다면 그 chatroom으로 이동시켜주고 없다면 create문을 통해서 만들어주는 작업을 거치게 됩니다. table명은 방금 얻은 10개의 길이를 가진 문자열과 방을 개설하고자 하는 사람의 아이디를 합쳐서 이름을 구성합니다. 만약에 숫자가 1489091919이고 개설하고자 하는 사람의 아이디가 hyeonjun이라면

create table 1489091919hyeonjun(

id char(12) not null,

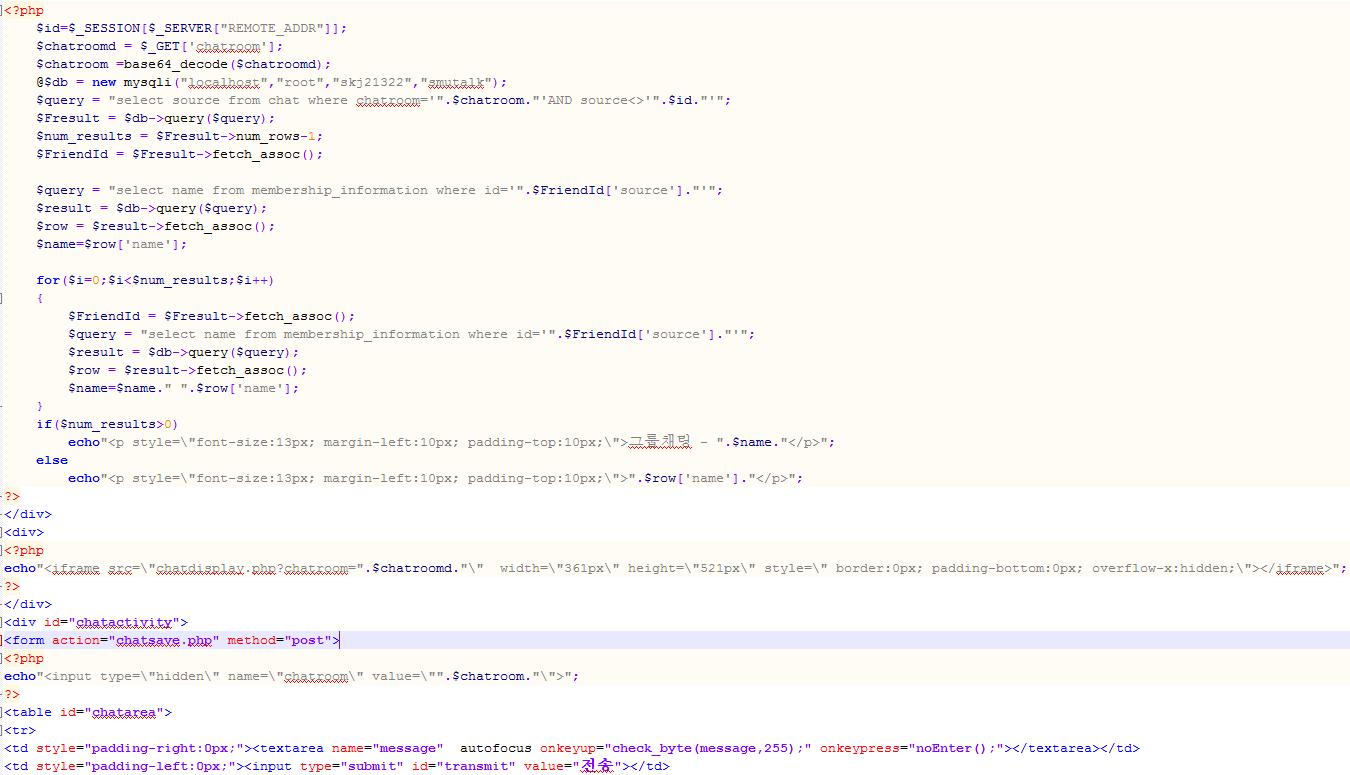
name char(40),

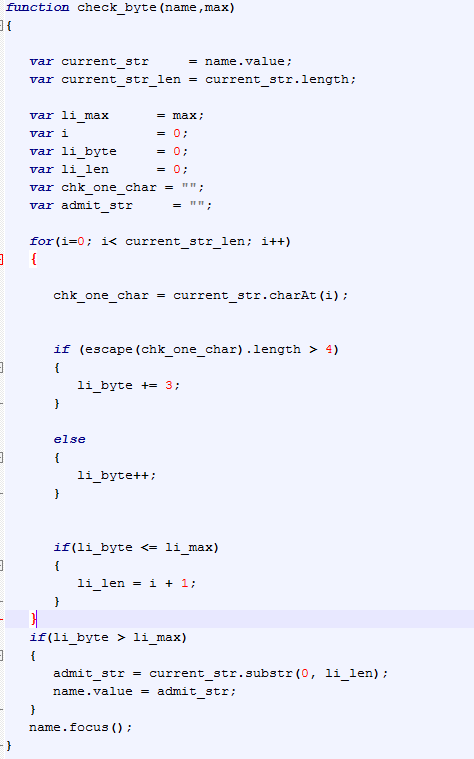
message char(255),

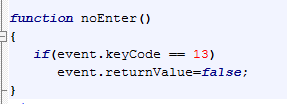
time char(10))Default charset=utf8; 이러한 구문을 통해서 만들어 지게 됩니다.

4개 이상의 인자를 가지고 있으면 group이기 때문에 먼저 선택한 대상들의 아이디를 가져오게 됩니다. 어떤 인덱스를 가지고 있는지 모르기 때문에 foreach문을 통해서 날짜와 채팅 방의 이름에 대한 내용을 뺀 나머지를 받아와서 base64\_decode시켜서 원래 아이디로 바꿔줍니다. 그룹 채팅 방의 경우에는 대상을 같은 사람으로 계속 선택해서 만들어도 다른 이름으로 채팅 방이 개설되는 특성을 가지고 있기 때문에, 언제 만들고 언제 이미 존재하는 그룹 방으로 이동시켜야 하는지 정해야 했습니다. 따라서 invite.php에서는 다수일 경우에 url에 빈 채팅 방의 이름을 넣어서 findchatRoom.php입장에서는 선택된 대상이 같더라도 계속해서 만들 수 있게 하였고, 나중에 나올 chatlist.php에서 존재하고 있는 chatroom을 더블 클릭하게 될 경우 chatroom이 이름이 있기 때문에 findchatroom.php에서는 이미 존재하는 chatroom로 이동시켜주도록 구성하기 위해서 이렇게 만들었습니다. 그룹 채팅을 생성할 때는 앞에서 1대1과 다른 점이 존재하게 되는데 1대1에서는 chatroom의 이름을 만들고 create을 통해서 채팅 내용을 저장할 수 있는 table을 구성한 후에 chat이라는 테이블에 insert문을 통해서 자신의 id와 대상의 id로 두 개의 레코드를 구성해서 넣어주게 됩니다. 이는 내가 송신자가 될 수도 있고, 대상이 송신자가 될 수 있기 때문에 두 개의 레코드를 구성한 것인데, 그룹일 경우에는 destination 즉, 수신자가 여러 명이기 때문에 대상들의 아이디를 적는 것이 아닌 g라는 문자를 저장하기로 했습니다. 그러면 나중에 같은 chatroom의 이름을 가지고 destination이 g를 가진 chatroom을 중복 없이 하나만 뽑아 내는 방법으로 다른 작업들을 쉽게 진행 할 수 있기 때문입니다.

-chat.php-







findchatRoom.php을 거치게 되면 chat.php로 이동하게 됩니다. 여기서는 인자로 받아온 chatroom으로 chat테이블에 chatroom컬럼이 chatroom의 값과 같은 레코드 중에서 자신source값이 자신의 아이디가 다른 사람들의 id를 가지고 옵니다. 만약에 chat.php가 그룹 채팅을 위해서 호출 된 것이라면 그 수가 다수가 될 것이며, 그러면 그룹채팅 part로 넘어갈 것이고, 1명이라면 1대1 part로 넘어갈 수 있게 if문을 구성하였습니다.

보통 메신저는 풀링의 작업을 거치게 되는데, 이것을 구성할 때 풀링 작업을 refresh로 대체 하였습니다. 따라서 chat.php를 refresh하게 되면 메시지 값을 입력 받을 수 없기 때문에 chatdisplay.php라는 다른 페이지를 <iframe>를 통해서 보여주고 chatdisplay.php를 refresh시켜줘서 풀링의 작업을 대신하도록 하였습니다.

해당 chatroom의 테이블에서는 message의 값이 255byte까지 허용이 가능합니다. 따라서 입력한 값이 255바이트까지 허용해야 하기 때문에 javascript 함수를 쓰게 되었습니다.

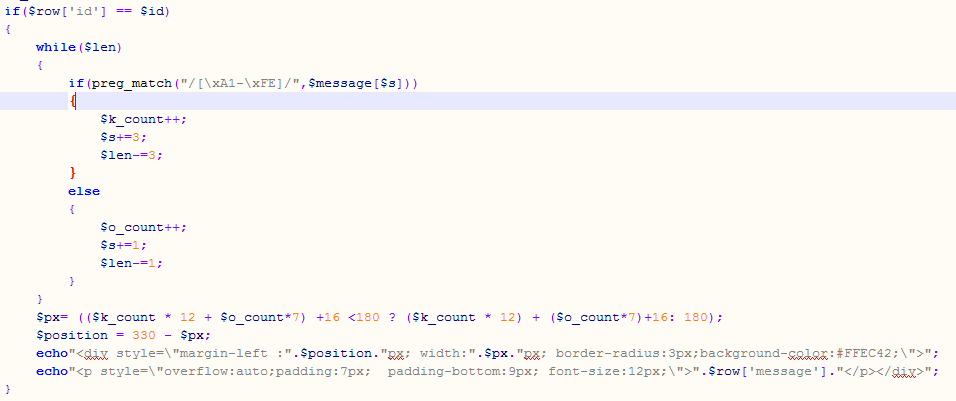
키보드 이벤트를 사용해서 utf8이기 때문에 한글은 3byte 나머지는 1byte로 계산해서 총합이 255byte가 넘게 되면 허용되는 문자열까지만 보여줌으로써 더 이상 입력 할 수 없다는 것을 보여주게 됩니다. 그 외에 Enter 값을 제외하기 위한 함수를 구성하였습니다.

-chatsave.php-



chat.php에서 전송버튼을 누르게 되면 chatsave.php에서 저장을 하게 됩니다. 이미 255byte까지 입력 할 수 있도록 검사가 진행 되어있기 때문에 해당 값이 비어있지 않다면 message를 보내는 사람의 id와 이름, 내용, 보내는 시간을 insert문을 통해서 해당 chatroom의 레코드로 삽입합니다.

-chatdisplay.php-





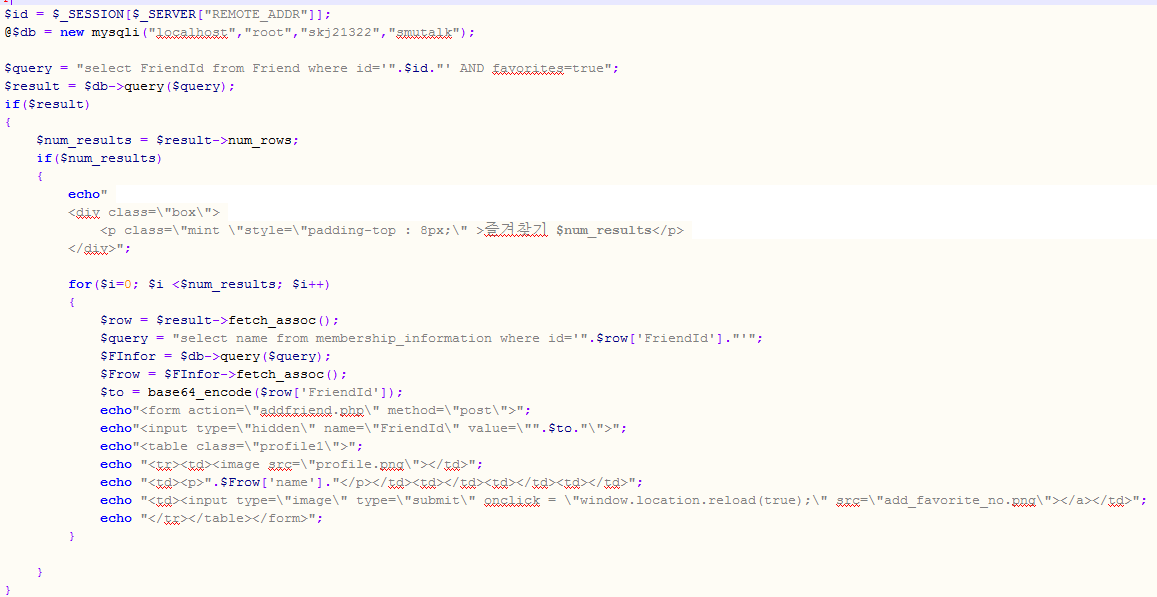
카카오 톡처럼 자신이 입력한 메시지의 배경색은 노란색으로, 상대방이 입력한 메시지는 기본 프로필사진 + 이름 + 메시지내용이 보일 수 있도록 구성하였습니다.

그래서 chatroom에서 레코드들을 받아온 후에 id가 자신의 id라면 if문을 실행, 자신의 id가 아니라면 else부분을 실행하도록 하였습니다.

if문과 else문의 공통점은 message의 내용을 받아와서 한글과 나머지의 수를 세고 한글의 수에 12를 곱하고 나머지에는 7을 곱한 후 16을 더해서 180px이상이면 180px까지를 제한으로 정해서 나머지 글자는 밑으로 계속 내려갈 수 있게 구성하였고, 180이하면 한 줄에 그 내용을 전부 표시해주는 형태로 구성하였습니다. 따라서 ‘네’ 혹은 ‘알겠습니다’ 의 message의 영역의 크기가 달라지는 효과를 볼 수 있습니다.

그 외에 if문은 배경색을 노란색으로 설정하고, else문은 프로필 사진과 이름 메시지 내용을 보여주기 위해서 <table>을 사용해서 구성 하였습니다.

-favorite.php-



Afterlogin.php에서 별 모양의 버튼을 누르게 되면 favorite.php로 이동하게 되는데, 친구들의 리스트를 보여주는 과정은 Afterlogin.php와 비슷하고, 다른 점은 <input>태그가 있어서 친구 프로필영역의 즐겨찾기 버튼을 클릭하게 되면 addfriend.php로 대상의 아이디를 전송하게 되고 favorite.php는 reload되어서 방금 즐겨찾기를 눌렀던 대상이 즐겨찾기가 되어있다는 의미로 친구 리스트의 상단에 즐겨찾기 리스트가 만들어져서 즐겨찾기가 된 대상이 나타나고, 즐겨찾기가 되지 않은 친구들의 리스트는 밑에 존재하게 됩니다.

-search.friend.php-



먼저 자신과 같은 학과인 사람들의 id를 받아오기 위해서 membership\_information에서 session에 저장 된 자신의 아이디로 학과를 받아오고, 다시 membership\_information에서 자신의 학과와 같은 사람들의 id를 받아옵니다. 그리고 자신이 친구 등록했던 사람들의 id를 받아오기 위해서 Friend테이블에 자신의 아이디로 등록한 친구들의 id를 받아옵니다. 바로 밑 for문에서 친구들의 id를 배열에 저장하고 다음 for문에서는 같은 학과의 사람들의 리스트들에서 이미 친구 등록한 사람을 제외하기 위해서 id가 같은지 비교하는 if문이 있습니다. 만약에 자신이 등록한 친구가 아니라면 <table>을 이용해서 프로필 영역을 만들어 줍니다.

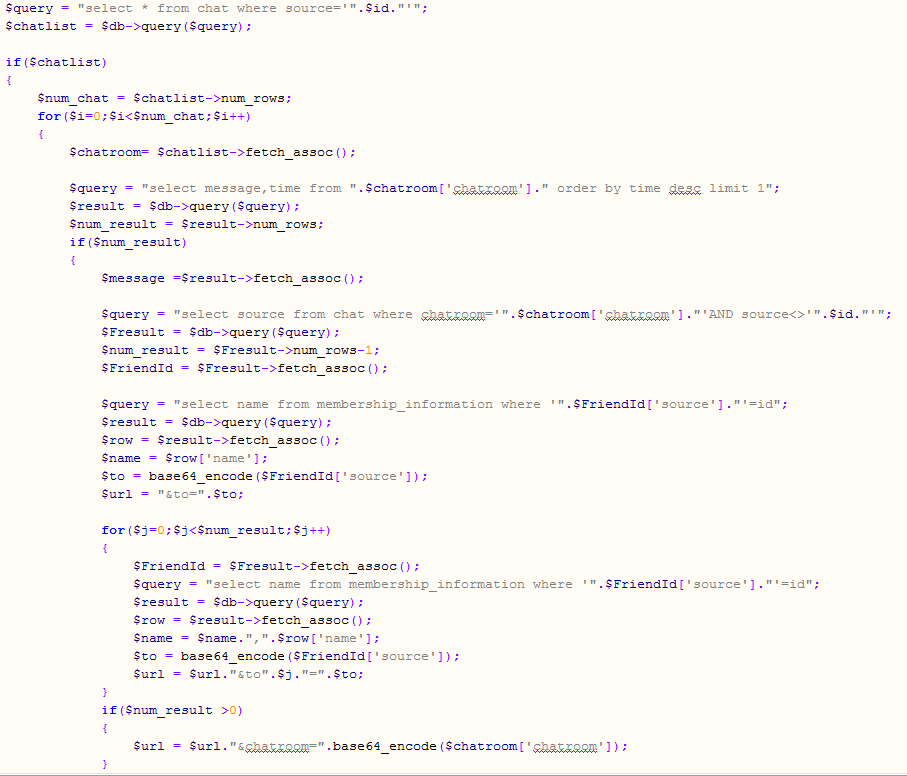
프로필 영역에서 친구 등록 버튼이 존재하게 되고 해당 버튼을 클릭하게 되면 위의 favorite.php와 같은 방식으로 진행하게 됩니다.

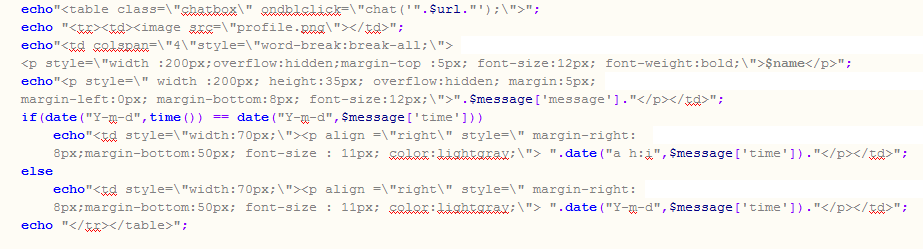
-addfriend.php-



favorite.php 또는 search.php에서 addfriend.php로 값을 넘겨주게 되는데, 먼저 Friend 테이블에서 자신의 아이디와 받은 친구의 id가 하나의 레코드 안에서 존재하는지의 여부를 판단합니다. 만약에 레코드가 있다면 favorite.php에서 addfriend.php로 이동해 온 것이기 때문에 Friend 테이블에서 값으로 넘겨받은 친구의 id가 존재하는 레코드의 컬럼 favorites가 true이면 false, false이면 true로 바꾸어 주는 작업을 거치게 되고, 레코드가 없다면 searchfriend.php에서 이동한 것이기 때문에 Friend 테이블에 자신의 id와 친구의 아이디, 그리고 favorites값은 false 즉 0으로 된 레코드를 insert해줍니다. 그리고 각 작업이 끝나게 되면 addfriend.php를 호출하기 전 페이지로 이동하게 됩니다.

-chatlist.php-





자신이 메시지를 보냈던 room들의 리스트를 보기 위해서 chat 테이블에서 자신의 아이디로 참여하고 있는 레코드들을 가지고 옵니다. 존재한다면 for문에서 레코드를 하나씩 fetech\_assoc()하면서 chatroom의 이름과, 참여하고 있는 사람들의 id를 가져와서 url을 구성해주고 <table>을 구성할 때 참여하고 있는 사람들의 이름 중 자신의 이름을 뺀 나머지를 표시해줍니다.

각 room들의 영역을 구성하는 <table>에는 더블 클릭하면 chat(url값)이 진행됩니다. afterlogin.php와 같은 방식으로 chat()함수는 동작합니다.

1. **마치면서**

-심현준-

Ajax를 썼으면 좀 더 완성도가 높아질 수 있었지만, 중간에 몇 번 실패하고 공부의 양이 부족했기 때문에 php와 javascript만으로 창의적으로 생각해서 진행해 보자고 결심 했었고, 진행 하는 도중에 여러 어려운 점들도 있었지만 몇몇 사람들에게 보여줬을 때 반응이 괜찮아서 성취도가 있었습니다. Ajax를 좀 더 공부할 시간이 생긴다면 제출하는 프로젝트에 멈추지 않고 개선하여 완성도를 높이는 계획도 생각해보고 있습니다.

-김기상-

많이 부족한 PHP 때문에 고전도 많이 했고 처음 해보는 웹 프로그래밍 때문에 고통도 많이 받았습니다. Css를 처음 접하는 나로써는 매우 생소 했다. 폴링 방법을 연구도 많이 했지만 생각만큼 진전이 되지 않았고 javascript는 정말로 공부의량이 적다고 생각이 되었습니다. 한 학기 동안 보람찬 프로젝트를 한 것 같습니다.