Project part II

Xin He N15648838

Hanzheng Wen N17492030

I At first

This is the first time for us to do team job separately. We both have different other job, the time left for us to work together is rare. So we can’t make this perfect. As times goes by and thanks to Google, we have been making progress smoothly, And we are stepping to perfect.

II System description

This is a website to provide a platform for both users and artists to get and share their information.

Users could get the information from the band that they like and from the recommend system. They could also make contribution to make the recommend system larger to make better service for more users.

Artists could provide new concert to public to inform the users who is his fan. In the future, we want to give the privilege to the artists to reduce the score of bad users to make the system better.

We use the database system from AWS as our backend and PHP + CSS as our front end.

III User Guide

Select User option at the start page and user will be transferred to user’s welcome page.

Users will use our welcome page to login or register.

If they need to log in, just type in the username and password. If the username and password match, then user will be lead to his homepage.

If they need to register, they just need to fill in their expecting username and password on the right bottom of the page, if the username haven’t been used yet, then it will led user to a new page to fill their personal information and upload his head sculpture.

On the top of homepage, user will see a navigation bar, which could lead user to six function pages.

The first one is the Home page. When user logged in successfully, they will first come to this page. This page does hold two parts, the first part is recommendation part, user could see the band and the user he might be interested in based on his tastes. The second part is all the artists and users on this site, which is designed for user to follow other user or band easily by one click.

The second one is the Artists page. User will see the artists they are following. Once they click on one artist, it will jump to the artist’s detail page. The description of this artist, the newest video and the concert he will attend will be shown up on this page.

The third one is Concert page. User could see all concerts that will be held in Upcoming Concert tab and the concert he will attend. Also, user could see the detail such as time, location and even the artist who will show up in this concert by clicking the concert image. If the concert is not full, user could RVSP to join this concert. In the concert detail page, which is led from clicking the concert image, also have a review function. If the user hasn’t been register for this concert, he could only see the review others left. But when the user shows his willingness to join this concert, he will be able to update his review left for this concert.

The forth page is used for showing lists. In this page, user could see the list he followed and the list he created. If the user didn’t create any list, then the My Lists tab will be invisible. There also have a create list option, which will lead user to a create list page, help user to create a list made up by concerts.

The fifth page is Connection page. User could see who is following him and whom he follows. User could see the detail of this user by click his photo, then a new page will come out showing this user’s personal detail.

The last page is users profile. Users could see his own profile like seeing others profile, but user could edit his own profile or add a new list in this page. User could edit his personal profile here. We also added create list function here so that user could create list easily.

IV Artist Guide

Select Artist option at the start page and artist will be transferred to artist’s welcome page.

Similar with User login page, artist could use this page to login or register. Use their own username or password to login or fill their information below to register for a new artist user account.

After logged in, there will be only one page here.

Artist could use this page to edit his profile or add new concert.

V Change of the backend System

We designed the backend System in the part I just based on requirement. We changed a lot about Database System constructer in this part.

First, we add Timestamp for users to record their register time and last login time, this could help server to trace their footstep, and further, calculate their score.

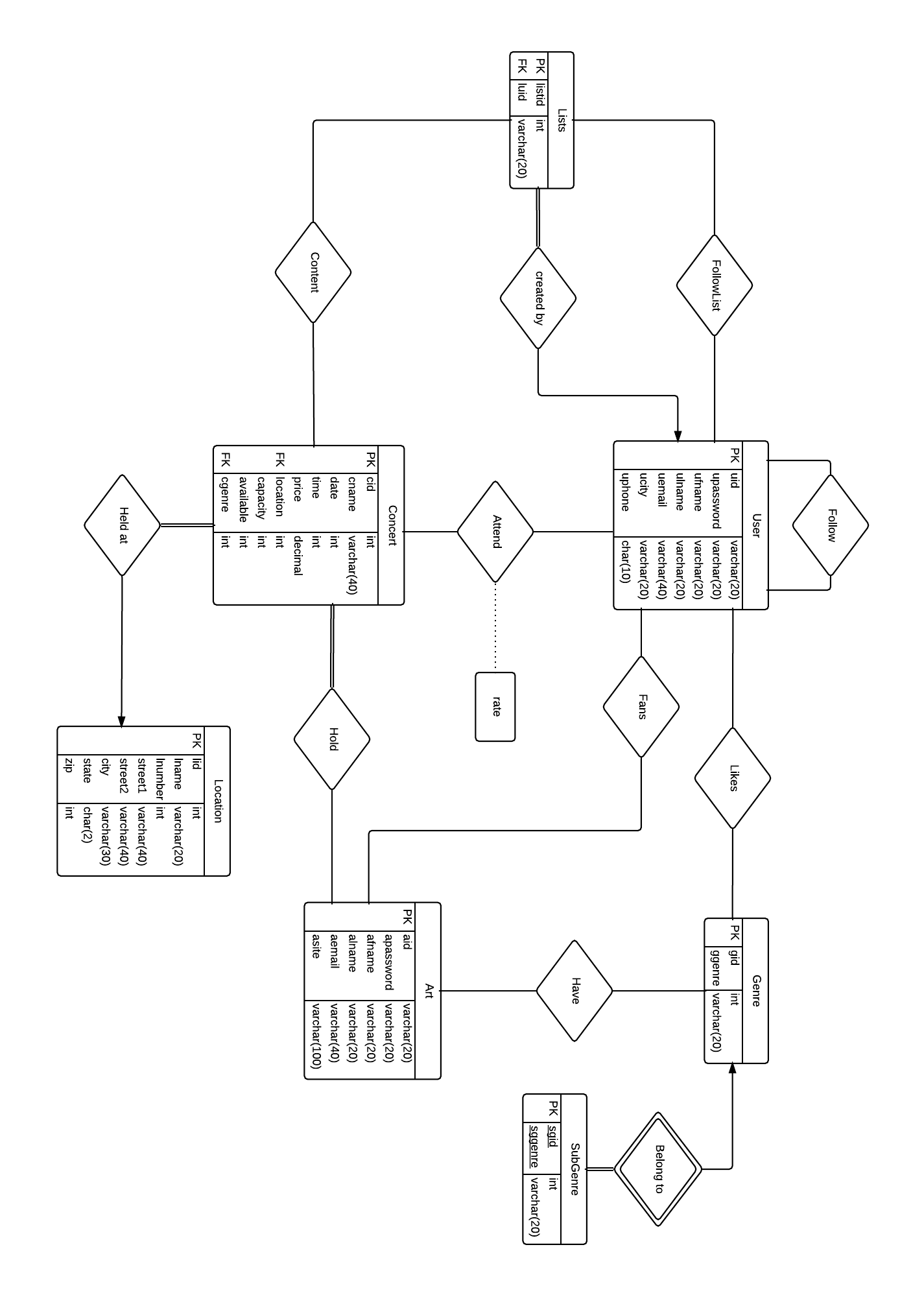
Then, we add bio attribute to store the description of user and artist himself (themselves) so that user could easily see what this band is about.

We also add attribute of self photo to users and artist. We just save a filename and save the photo of these user in the photo folder. In this way, we just need to take care about the filename instead of the metadata and give the job of showing photo to front end.

What’s more, the YouTube page is added to artists table so that users could see their latest video directly.

At last, and the most important part, we add number attribute to user, artist and set all the entity defined with number have the contribution of self auto increment so that we don’t need to care about the id of new user.

VI E-R diagram



VII Explanations of diagram

1. User

This is almost the most import part in the whole system, because more than half of the functions have something to do with User. We create the table by

create table User (

uid int(10) AUTO\_INCREMENT,

uid varchar(20),

upassword varchar(20),

ufname varchar(20),

ulname varchar(20),

uemail varchar(40),

ucity varchar(20),

uphone varchar(10),

regtime datetime,

lastlogin datetime,

ulink varchar(200),

ubio varchar(500),

uscore int(3),

primary key (uid)

);

uid, which is defined as username, is the primary key to distinguish different user. upassword is to save user password, will be encrypted in front end and save encrypted password as plain text here. ufname, ulname, uemail, ucity and uphone is used to save users’ inforamation. regtime will be used to save the users register time, this attribute will be used for updating uscore, which is used to judge the score of user, later.

2.Arts

Similar with User, we create the table like this.

create table Art (

aid int(10) AUTO\_INCREMENT,

auname varchar(20),

apassword varchar(20),

artname varchar(20),

aemail varchar(40),

asite varchar(100),

alink varchar(200),

abio varchar(500),

youtube varchar(45),

primary key (aid)

);

Arts also have user name as aid, and password as apassword. And we define aid as primary key. But we store the artists’ information as artname, which is bands’ or singers’ name, aemail and asite. Here, we didn’t save the regtime of any time stamp for singer, but if we need it, we will add them later.

3. Location

This is an entity which is solid and independent. So, we create this table as the third place.

create table Location(

lid int(10) AUTO\_INCREMENT,

lname char(20),

lnumber int(8),

street1 varchar(40),

street2 varchar(40),

city varchar(30),

state char(2),

zip int(5),

google varchar(100),

primary key (lid)

);

Different location will be given different lid as primary key, the lname is to store the name of the place. The other attributes are to store the address of this location, particularly, google attribute is used for saving google map location.

4. Concert

Concerts only have something to do with location and arts, so we decided to build Concert here as follow.

create table Concert (

cid int(10),

cname varchar(40),

holdtime datetime,

price decimal(7,2),

location int(10),

capacity int(6),

available int(6),

clink varchar(200),

cbio varchar(500),

primary key (cid),

foreign key (location) references Location(lid)

);

different concert have different cid which will be given by backend as primary key, and location is references from lid of location, then other attributes will be about this concert.

5. Hold

Since the relation between concerts and arts is many-to-many relation, then we need table to save the relation between concerts and artists as hold

create table Hold (

haid varchar(20),

hcid int(20),

foreign key (haid) references Art(aid),

foreign key (hcid) references Concert(cid)

);

We could easily find all the concerts artists attend and all the artists every concert have.

6.Genre and SubGenre

Genre is to define what kind of style the user like or what kind of style the artists belong. SubGenre is a weak entity that belongs to Genre, which will be useful to define the category of artists. We created these two tables as follow.

create table Genre (

gid int(2) AUTO\_INCREMENT,

ggenre varchar(20),

primary key (gid)

);

create table SubGenre(

sgid int(4) AUTO\_INCREMENT,

ggid int(2),

sggenre varchar(20)，

primary key (sgid, ggid),

foreign key (ggid) references Genre(gid)

);

Gernre is the style of a category, we connect all the information about user’s taste with subgenre.

7. Likes

this is a many-to-many relation between users and Genres to describe what kind of genre the user like.

create table Likes (

luid int(10),

lgenre int(2),

lsgenre int(2),

foreign key (luid) references User(uid),

foreign key (lgenre) references Genre(gid)

);

8. Have

This relation is to define what style the artists belong to. Since one artist may have different kinds of genre, then we need to have a many-to-many relation to define this relation and create a table as follow.

create table Have (

haid int(10),

hgenre int(2),

hsgenre int(2),

foreign key (haid) references Art(aid),

foreign key (hgenre) references Genre(gid)

);

9. Follow

We need a relation between one user to others users as follow relation, we will use this relation to finish one way of recommendation, and even use this realtion to calculate users score later. So we build a relation as follow.

create table Follow (

followee varchar(20),

follower varchar(20),

foltime datetime,

foreign key (follower) references User(uid),

foreign key (followee) references User(uid)

);

10. Fans

We also need a relation between users and artists as fans so that users would get notified about news from these artists. The relation is as follow

create table Fans (

fan varchar(20),

follow varchar(20),

fantime datetime,

foreign key (fan) references User(uid),

foreign key (follow) references Art(aid)

);

11. Lists

every user could build lists that they like to recommend others to attend, in this way

create table Lists (

listid int(10) AUTO\_INCREMENT,

luid varchar(20),

creatime datetime,

primary key (listid),

foreign key (luid) references User(uid)

);

Every list would be given a listid as its primary key. luid is the attribute to mark the creator of this list and creatime is the time this list is created, we could use this stamp to notify others who follow this list.

12. Content

Every list could contain many concerts, and every concets could be contained in many lists, in this way, we build a table to describe this relation as follow.

create table Content (

clistid int(10),

ccid int(10),

primary key (clistid, ccid)

foreign key (clistid) references Lists(listid),

foreign key (ccid) references Concert(cid)

);

13. FollowList

Before Users follow other Users, they should follow the list they like before, So we need a relation between users and lists as followlist.

create table FollowList (

flistid int(10),

fluid varchar(20),

folltime datetime,

primary key (flistid, fluid),

foreign key (flistid) references Lists(listid),

foreign key (fluid) references User(uid)

);

14. Attend

This is an important part of the website, this table is used by users to show their willing of plan to attend some concerts, and later to store the review from these users. We build this table as follow

create table Attend (

auid varchar(20),

acid int(10),

rate int(1),

review varchar(500),

reviewtime datetime,

foreign key (auid) references User(uid),

foreign key (acid) references Concert(cid)

);

the rate attribute is for user to rate the concert, the review is their review of the concert, then the reviewtime is to show their time when they make the review.

V. Simple Test case

We inserted some data to check the SQL sentences as follow.

-- User

insert into user values('aaa', '123456', 'Peter', 'Allen', 'P.Allen@gmail.com', 'New York', '6461234567');

insert into user values('tomlee', '654321', 'Tom', 'Lee', ‘tomlee@yahoo.com', ‘New York', ‘7181234567');

insert into user values('lovemusic', ‘lovemusic', ‘Alice', ‘Julia', ‘lovemusic@gmail.com', ‘Brooklyn', ‘7185455678');

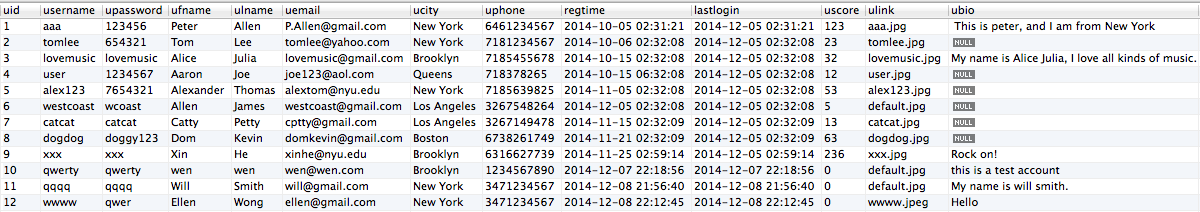
insert into user values('user', ‘1234567', ‘Aaron', ‘Joe', ‘joe123@aol.com', ‘Queens', ‘718378265');

insert into user values('alex123', ‘7654321', ‘Alexander', ‘Thomas', ‘alextom@nyu.edu', ‘New York', ‘7185639825');

insert into user values('westcoast', ‘wcoast', ‘Allen', ‘James', ‘westcoast@gmail.com', ‘Los Angeles', ‘3267548264');

insert into user values('catcat', ‘catcat', ‘Catty', ‘Petty', ‘cptty@gmail.com', ‘Los Angeles', ‘3267149478');

insert into user values('dogdog', ‘doggy123', ‘Dom', ‘Kevin', ‘domkevin@gmail.com', ‘Boston', ‘6738261749');



-- Arts

insert into art values('linkinpark', ‘12345678', ‘Linkin Park', ‘admin@linkinpark.com', ‘www.linkinpark.com');

insert into art values('jblunt', ‘12345678', ‘James Blunt', ‘jblant@gmail.com', ‘www.jamesblunt.com');

insert into art values('lenka', ‘12345678', ‘Lenka', ‘lenka@sina.com', ‘www.lenkamusic.com');

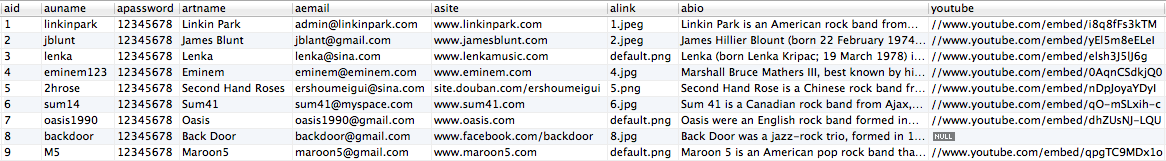
insert into art values('eminem123', ‘12345678', ‘Eminem', ‘eminem@eminem.com', ‘www.eminem.com');

insert into art values('2hrose', ‘12345678', ‘Second Hand Roses', ‘ershoumeigui@sina.com', ‘http://site.douban.com/ershoumeigui/');

insert into art values('sum14', ‘12345678', ‘Sum41', ‘sum41@myspace.com', ‘www.sum41.com');

insert into art values('oasis1990', ‘12345678', ‘Oasis', ‘oasis1990@gmail.com', ‘www.oasis.com');

insert into art values('backdoor', ‘12345678', ‘Back Door', ‘backdoor@gmail.com', ‘www.facebbok.com/backdoor');



-- Location

insert into location values('9001', ‘Barkley Center',620, ‘Atlantic Avenue',NULL, ‘Brooklyn', ‘NY', ‘11217');

insert into location values('9002', ‘The Way Station',683, ‘Washington Ave',NULL, ‘Brooklyn', ‘NY', ‘11238');

insert into location values('9003', ‘Barbès',376, ‘9th St',NULL, ‘Brooklyn', ‘NY', ‘11215');

insert into location values('9004', ‘Fat Cat',75, ‘Christopher St',NULL, ‘New York', ‘NY', ‘10014');

insert into location values('9005', ‘Bar Nine',807, ‘9th Ave',NULL, ‘New York', ‘NY', ‘10019');

insert into location values('9006', ‘Paddy Reillys',519, ‘2nd Ave',NULL, ‘New York', ‘NY', ‘10016');

insert into location values('9007', ‘Terraza 7',4019, ‘Gleane St',NULL, ‘Queens', ‘NY', ‘11373');

insert into location values('9008', ‘Blue Whale Bar',123, ‘Astronaut E S Onizuka St',NULL, ‘Los Angeles', ‘CA', ‘90012');



-- Concert

insert into concert values(100001, ‘SumSum', ‘2013-10-01 21:00:00',99,9002,300,null);

insert into concert values(100002, ‘new seccond hand', '2013-11-11 18:00:00', 20, 9007, 150, null);

insert into concert values(100003, ‘piece green', '2013-12-12 14:00:00', 65, 9006, 500, null);

insert into concert values(100004, ‘live in newyork', '2013-12-19 15:00:00', 120, 9001, 15000, null);

insert into concert values(100005, ‘newyear lenka', '2013-12-31 21:00:00', 80, 9004, 200, null);

insert into concert values(100006, ‘brait music', '2014-03-21 19:15:00', 40, 9005, 150, null);

insert into concert values(100007, ‘music aint junk', '2014-05-01 23:00:00', 25, 9007, 160, null);

insert into concert values(100008, ‘open the door', '2014-07-11 20:30:00', 15, 9006, 200, null);

insert into concert values(100009, ‘not alone', '2014-11-11 11:11:11', 11, 9002, 200, null);

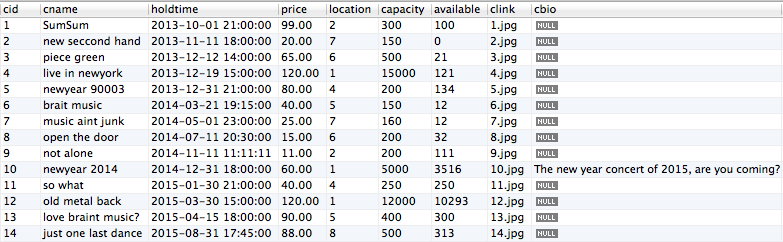
insert into concert values(100010, ‘newyear lenka 2014', '2014-12-31 18:00:00', 60, 9001, 5000, null);

insert into concert values(100011, ‘so what', '2015-01-30 21:00:00', 40, 9004, 250, null);

insert into concert values(100012, ‘old metal back', '2015-03-30 15:00:00', 120, 9001, 12000, null);

insert into concert values(100013, ‘love braint music?', '2015-04-15 18:00:00', 90, 9005, 300, null);

insert into concert values(100014, ‘just one last dance', '2015-08-31 17:45:00', 88, 9008, 500, null);



-- Hold

insert into Hold values('sum14',100001);

insert into Hold values('2hrose',100002);

insert into Hold values('oasis1990',100003);

insert into Hold values('linkinpark',100004);

insert into Hold values('2hrose',100004);

insert into Hold values('lenka',100005);

insert into Hold values('jblunt',100006);

insert into Hold values('eminem123',100007);

insert into Hold values('backdoor',100008);

insert into Hold values('2hrose',100009);

insert into Hold values('lenka',100010);

insert into Hold values('eminem123',100011);

insert into Hold values('sum14',100011);

insert into Hold values('linkinpark',100012);

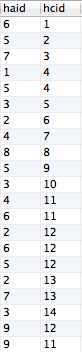
insert into Hold values('sum14',100012);

insert into Hold values('2hrose',100012);

insert into Hold values('jblunt',100013);

insert into Hold values('oasis1990',100013);

insert into Hold values('lenka',100014);



-- Genre gid ggenre

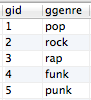
insert into Genre values(1, ‘pop');

insert into Genre values(2, ‘rock');

insert into Genre values(3, ‘rap');

insert into Genre values(4, ‘funk');

insert into Genre values(5, ‘punk');



-- Likes

insert into Likes values('aaa',1);

insert into Likes values('aaa',5);

insert into Likes values('alex123',2);

insert into Likes values('alex123',5);

insert into Likes values('catcat',1);

insert into Likes values('catcat',3);

insert into Likes values('catcat',4);

insert into Likes values('dogdog',1);

insert into Likes values('lovemusic',5);

insert into Likes values('tomlee',2);

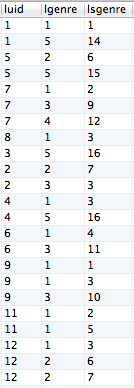
insert into Likes values('tomlee',3);

insert into Likes values('user',1);

insert into Likes values('user',5);

insert into Likes values('westcoast',1);

insert into Likes values('westcoast',3);



-- Have

insert into Have values(5,2,6);

insert into Have values(8,2,7);

insert into Have values(8,3,9);

insert into Have values(4,1,1);

insert into Have values(4,3,10);

insert into Have values(2,1,2);

insert into Have values(3,1,3);

insert into Have values(1,2,6);

insert into Have values(1,1,3);

insert into Have values(7,1.5);

insert into Have values(7,2,7);

insert into Have values(6,2,7);

insert into Have values(6,5,7);



-- Follow

insert into Follow values(1,9,'2014-01-01 15:37:29');

insert into Follow values(2,9,'2014-01-01 15:38:29');

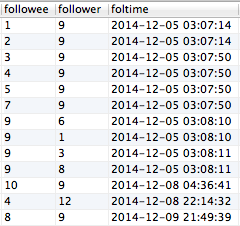
insert into Follow values(3,9,'2014-01-01 15:39:29');

insert into Follow values(4,9,'2014-10-01 15:39:29');

insert into Follow values(5,9,'2014-10-02 15:39:29');

insert into Follow values(7,9,'2014-11-02 15:39:29');

insert into Follow values(9,6,'2014-11-08 15:39:29');



-- Fans

insert into Fans values(9,5,'2014-01-01 15:37:29');

insert into Fans values(9,8,'2014-01-01 15:38:29');

insert into Fans values(9,4,'2014-01-01 15:39:29');

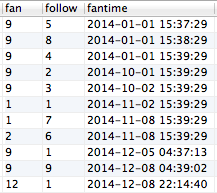
insert into Fans values(9,2,'2014-10-01 15:39:29');

insert into Fans values(9,3,'2014-10-02 15:39:29');

insert into Fans values(1,1,'2014-11-02 15:39:29');

insert into Fans values(1,7,'2014-11-08 15:39:29');

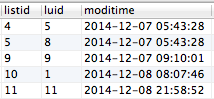
insert into Fans values(2,6,'2014-11-08 15:39:29');



-- Lists

insert into Lists values(6001,'alex123',NULL);

insert into Lists values(6002, 'dogdog',NULL);



-- Content

insert into Content values(4, 1);

insert into Content values(4, 11);

insert into Content values(4, 4);

insert into Content values(4, 5);

insert into Content values(5, 2);

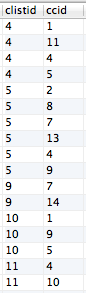
insert into Content values(5, 8);

insert into Content values(5, 7);

insert into Content values(5, 13);

insert into Content values(5, 4);

insert into Content values(5, 9);



-- FollowList

insert into FollowList values(4,5,'2014-10-01 18:54:39');

insert into FollowList values(4,7,'2014-05-09 09:09:55');

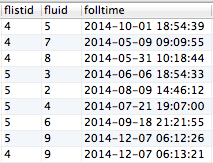
insert into FollowList values(4,8,'2014-05-31 10:18:44');

insert into FollowList values(5,3,'2014-06-06 18:54:33');

insert into FollowList values(5,2,'2014-08-09 14:46:12');

insert into FollowList values(5,4,'2014-07-21 19:07:00');

insert into FollowList values(5,6,'2014-09-18 21:21:55');



-- Attend

insert into Attend values(1, 1, 5, null, null);

insert into Attend values(5, 1, null, null, null);

insert into Attend values(7,10, 4, 'perfect', '2014-04-04 19:31:31');

insert into Attend values(8,8, 4, 'This is the best perform I have ever attend', '2014-05-05 18:41:55');

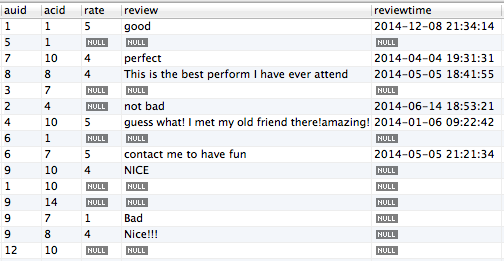
insert into Attend values(3,7,null,null,NULL);

insert into Attend values(2,4, null, 'not bad', '2014-06-14 18:53:21');

insert into Attend values(4, 10,5, 'guess what! I met my old friend there!amazing!', '2014-01-06 09:22:42');

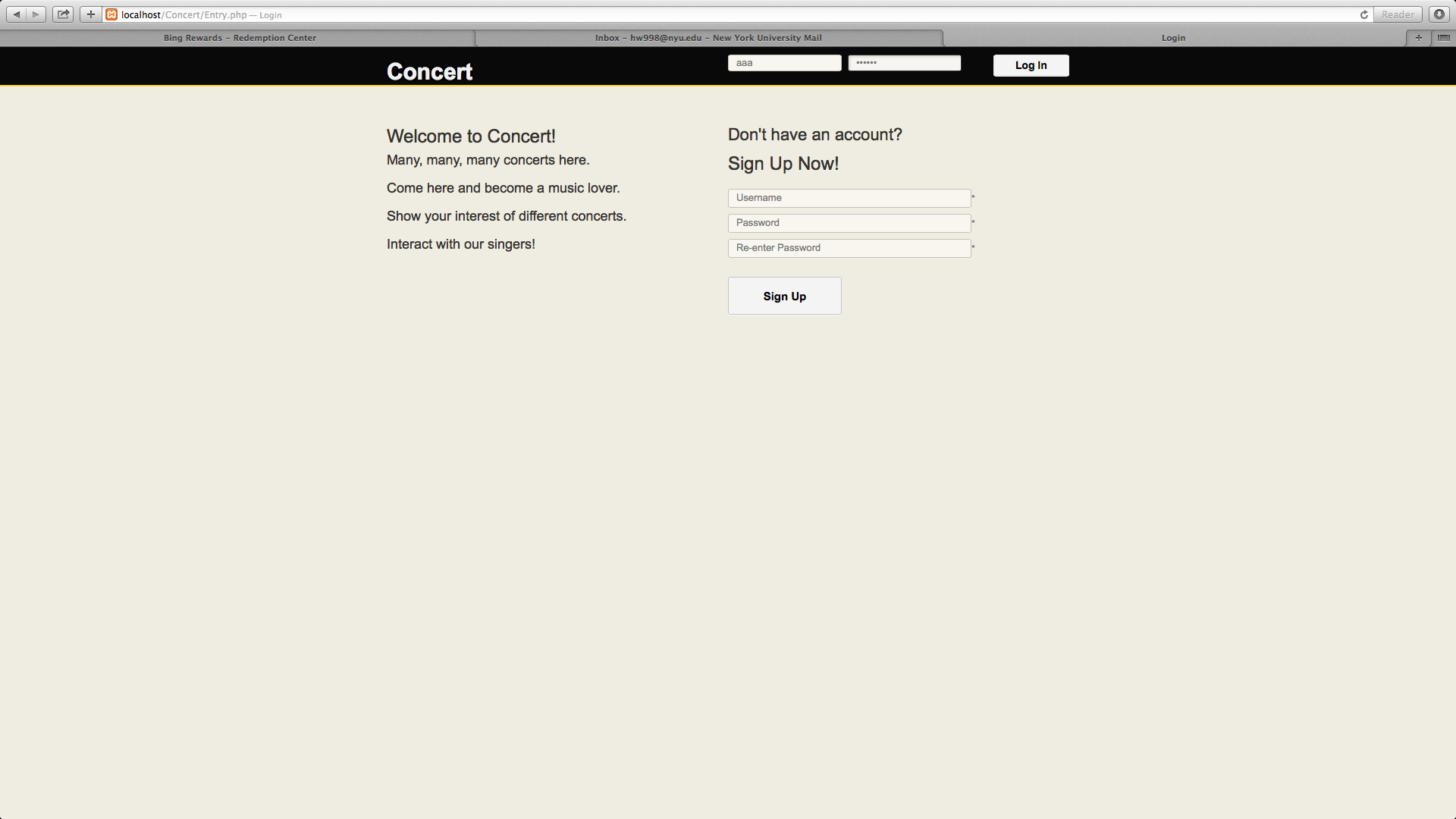
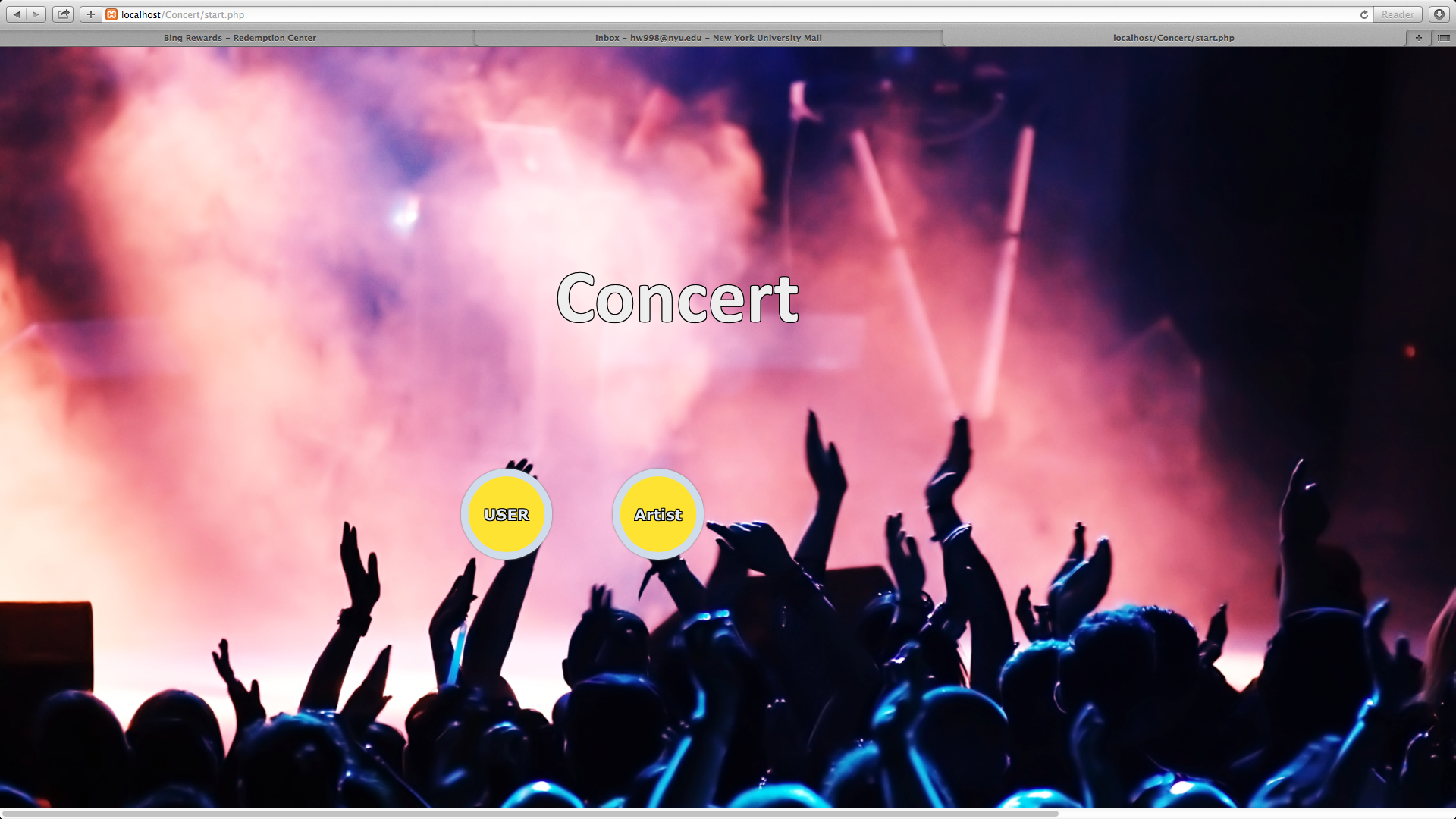
insert into Attend values(6,1, null, null, null);

insert into Attend values(6, 7, 5, 'contact me to have fun', '2014-05-05 21:21:34');

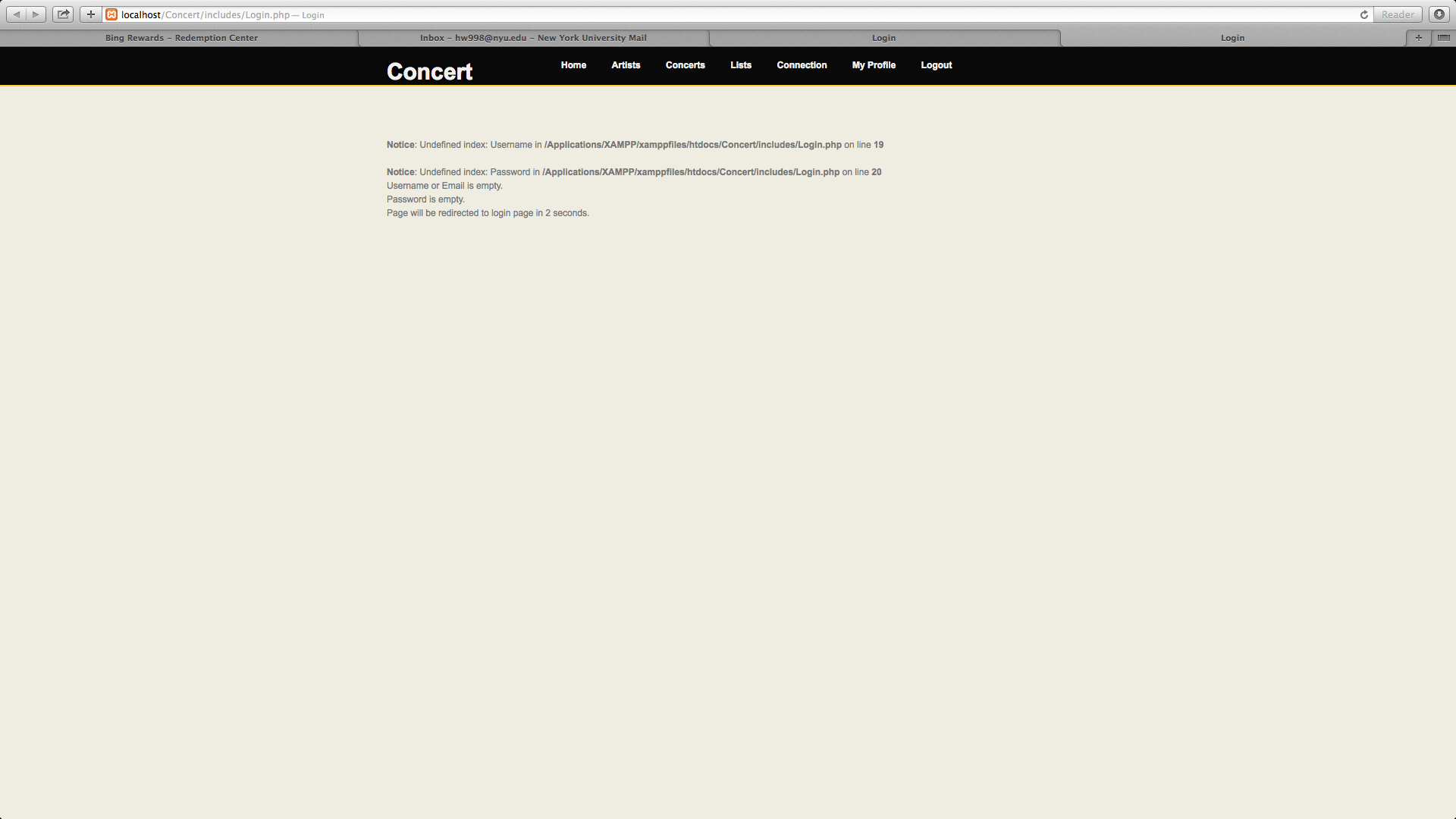


VI Some function test and result based on testcase

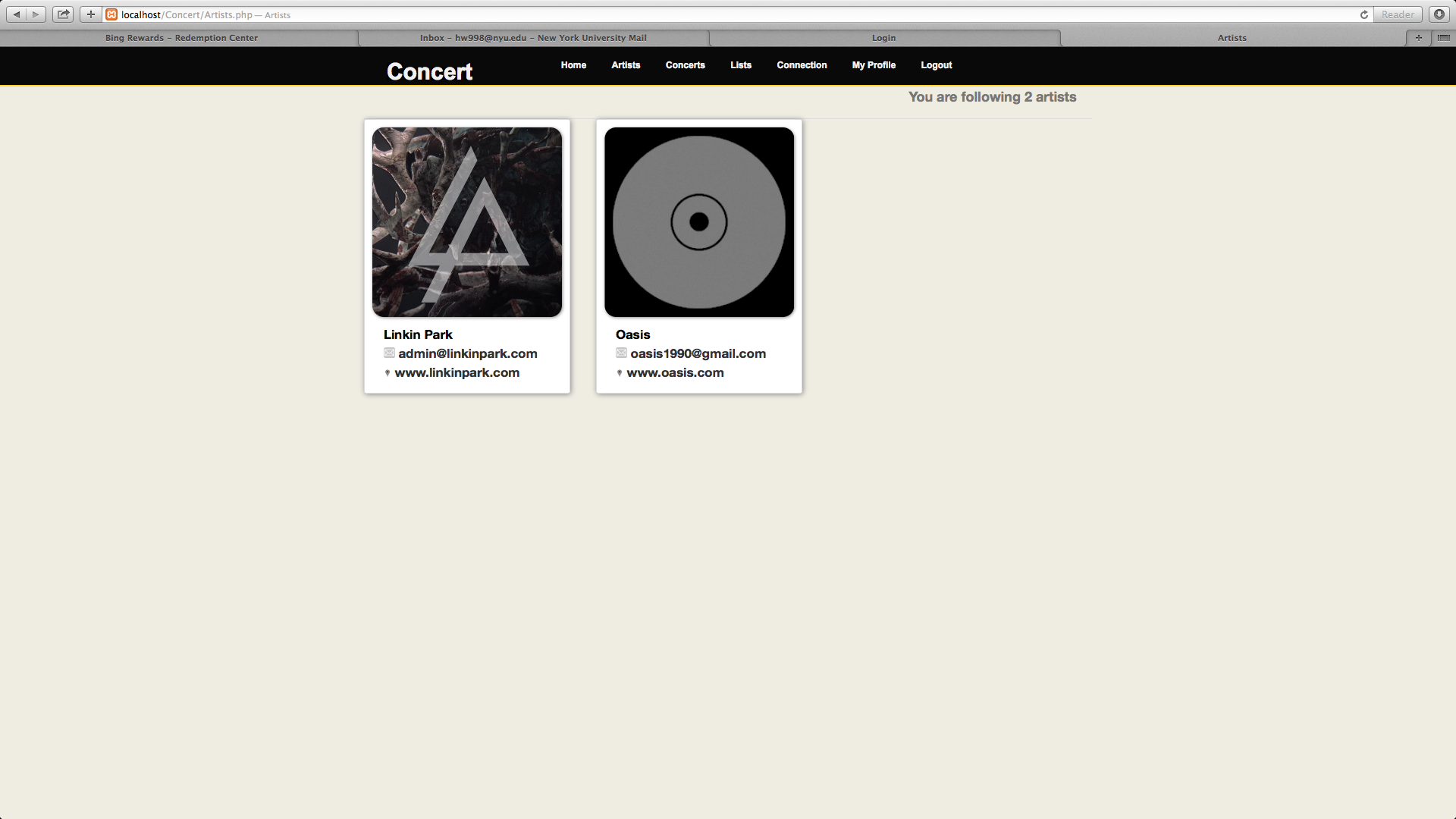
1. user login with right username w. password and wrong username w. password



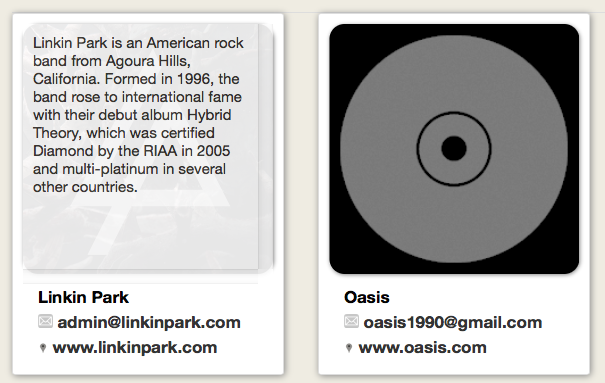
wrong password and right password:



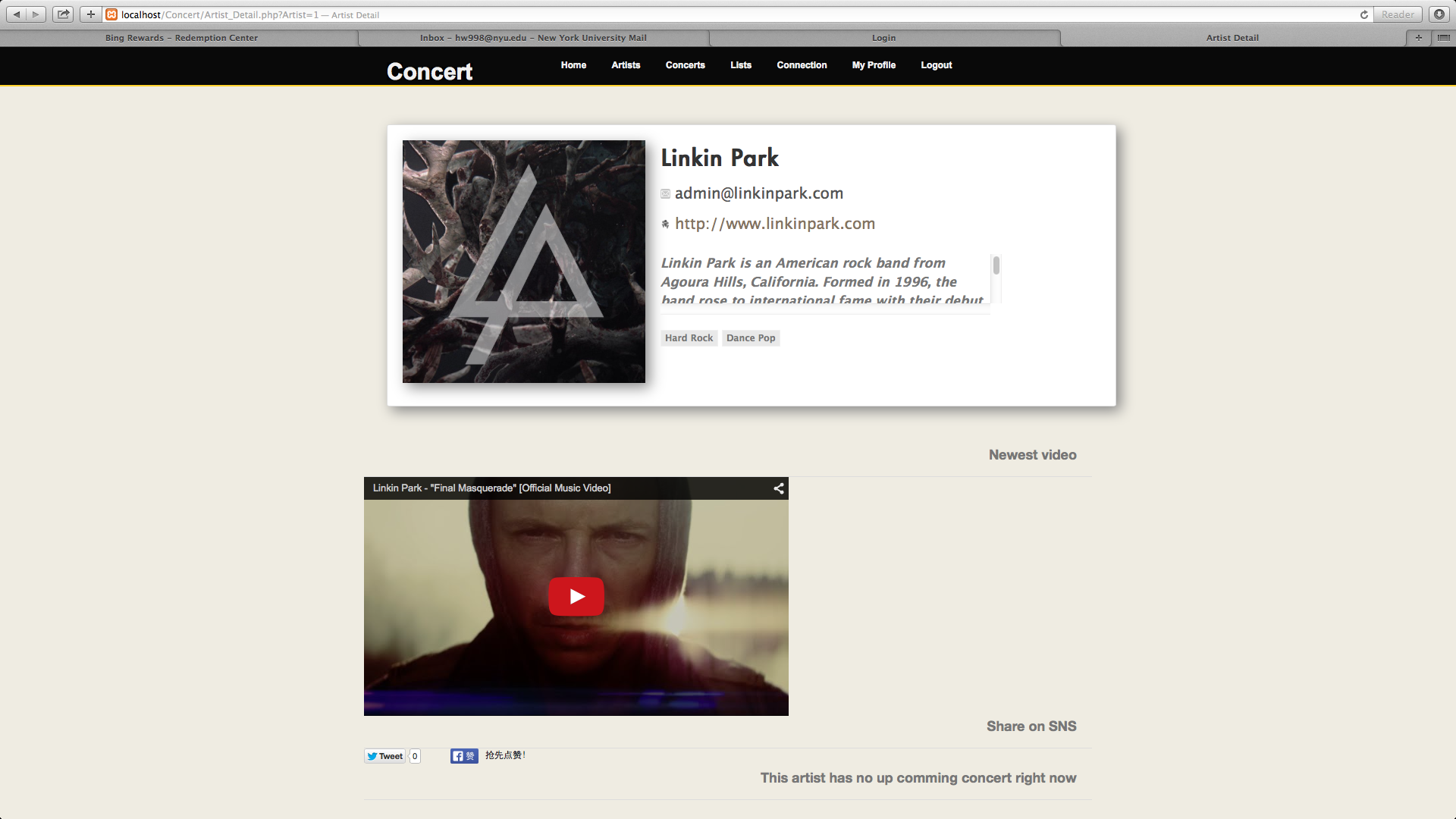
2. user’s following artist list page



user could see artist’s detail by moving mouse to artist location



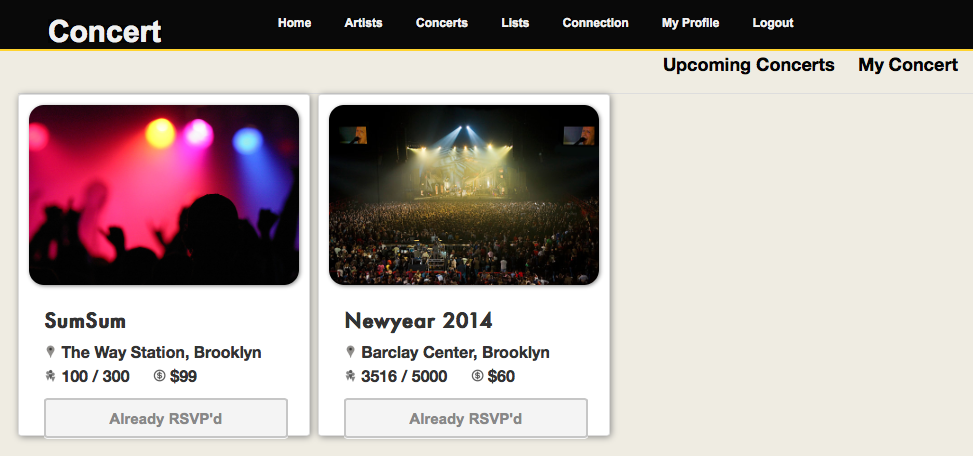
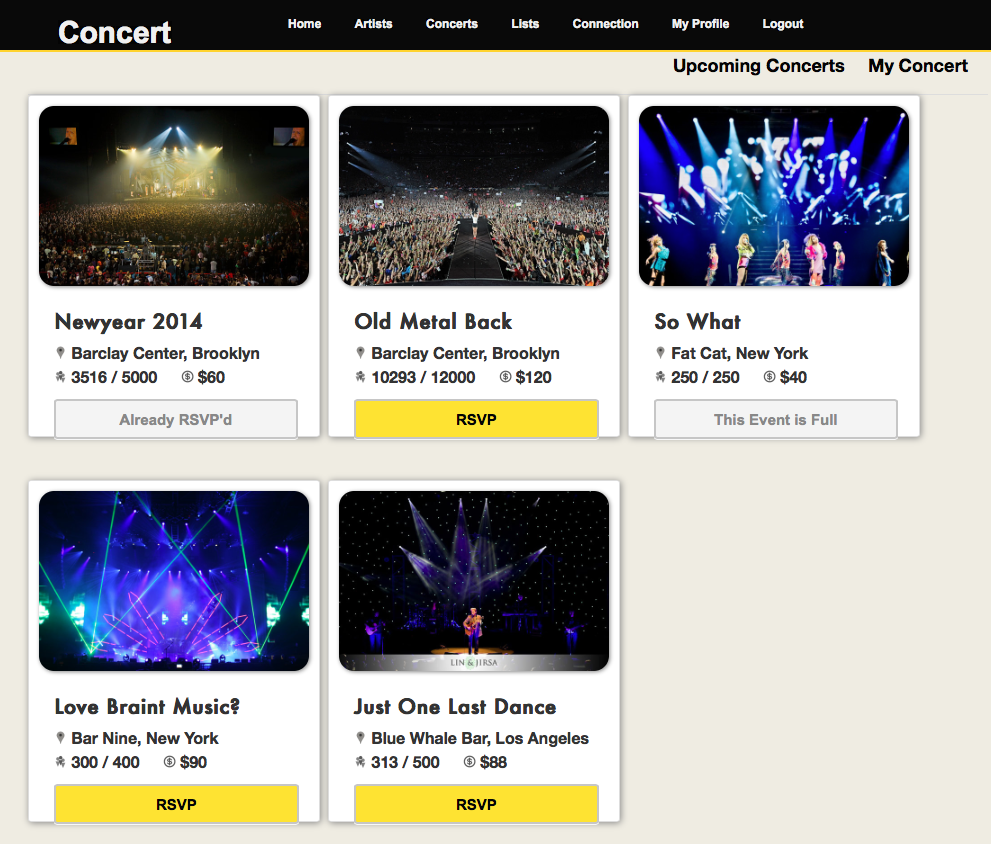
3.see the detail of an artist



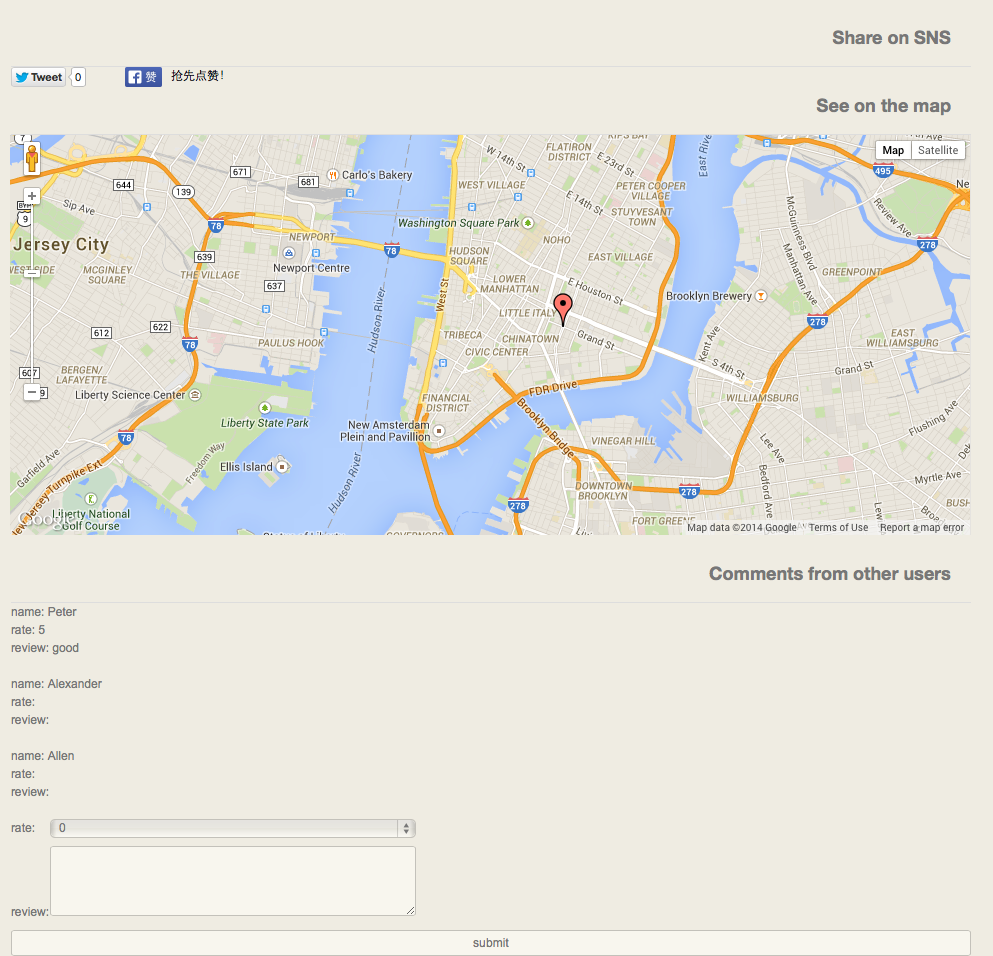
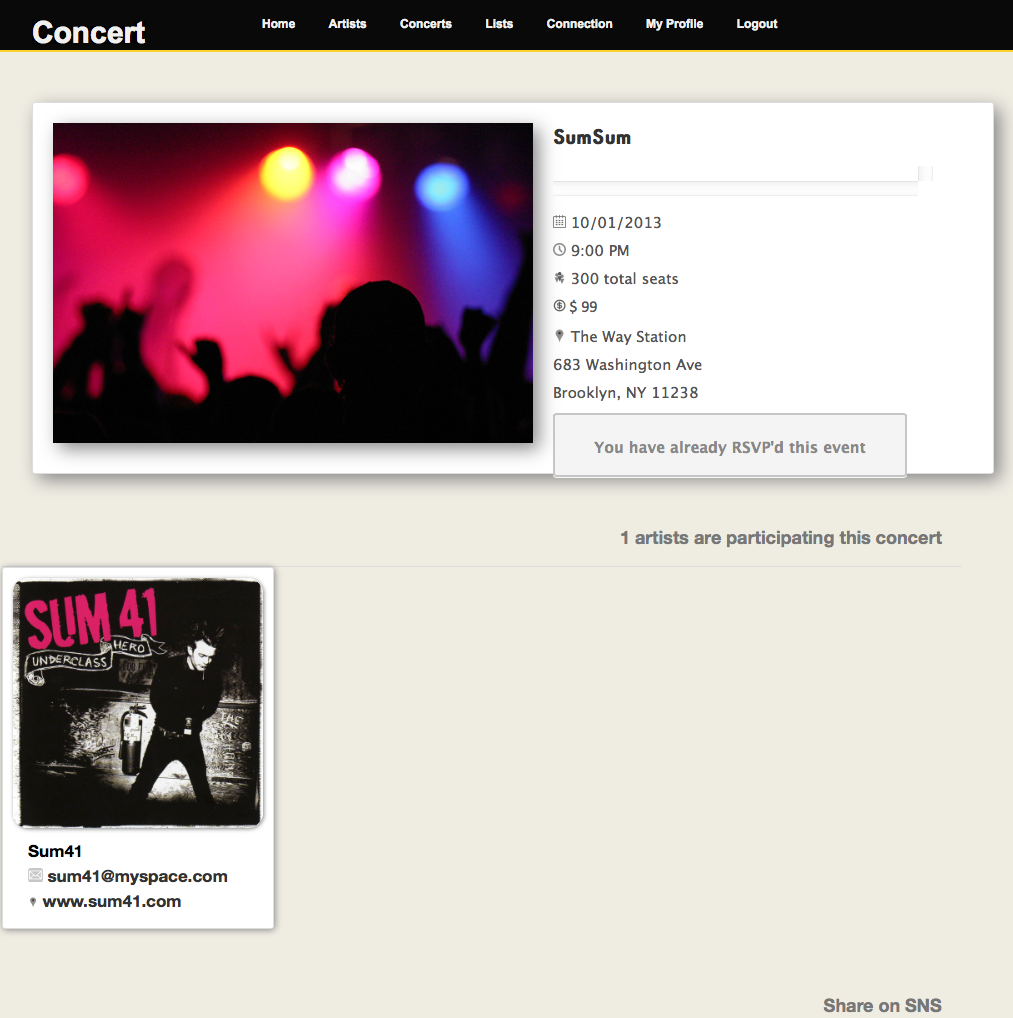
we could click play to play the youtube video and share this page on facebook and twitter:



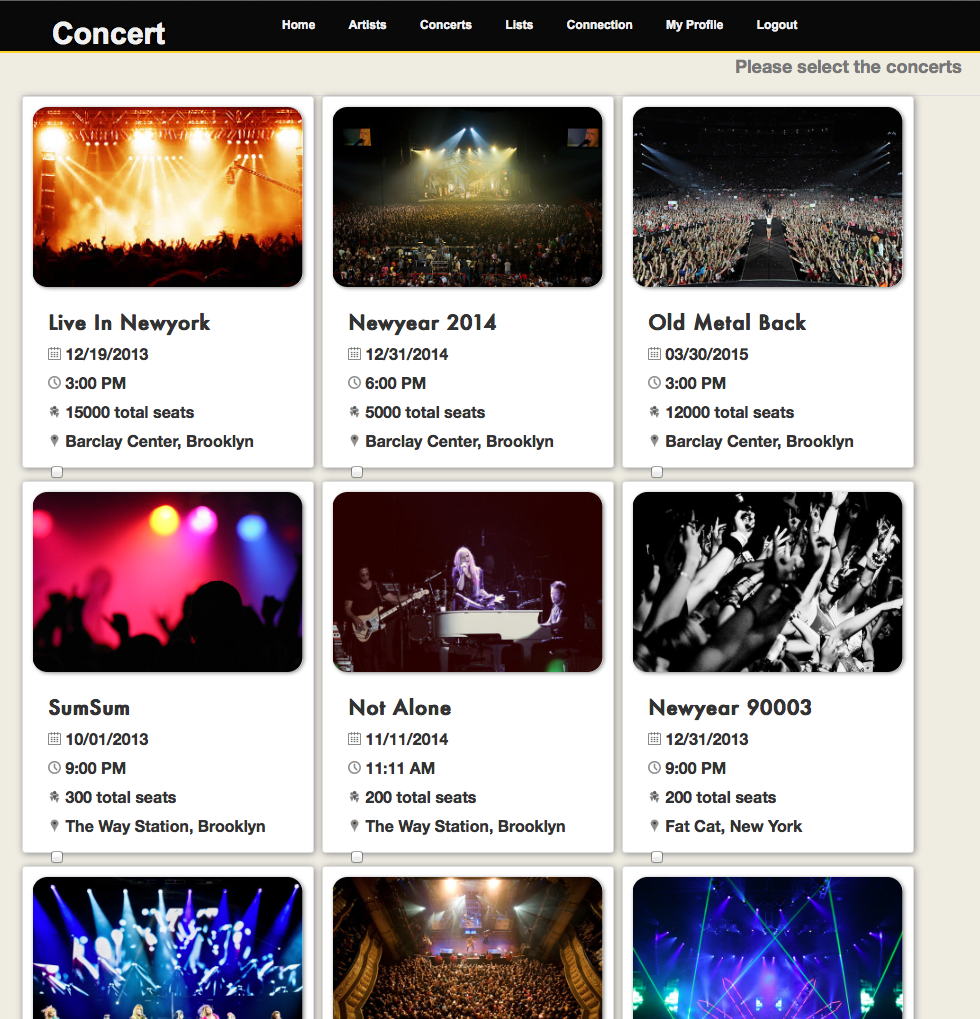
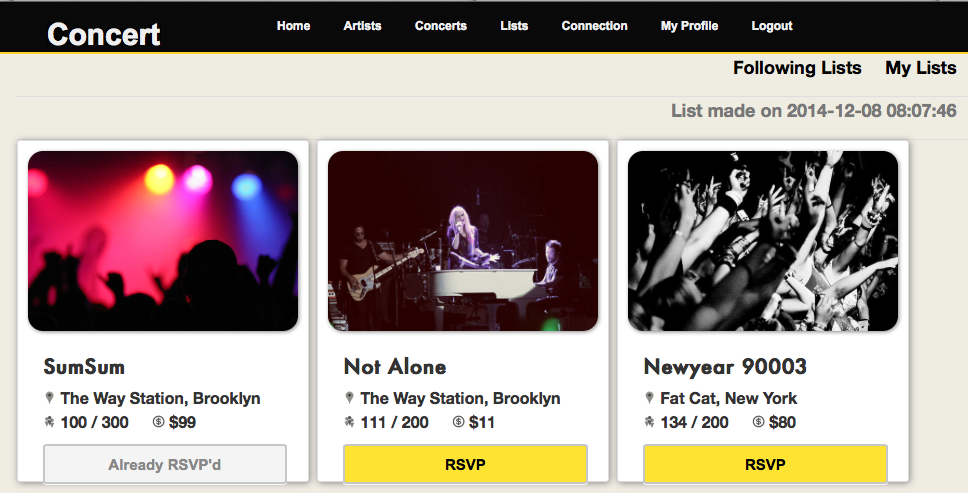
4. We could see the upcoming concert or my attended concert



5.In the concert detail page, we could see the artist coming to the concert location, also, we could share this page to twitter & facebook and update out comment here.



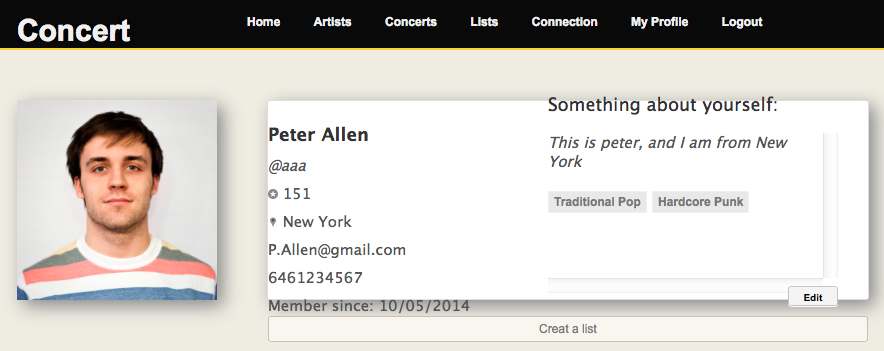
6. In the list page, we could see our following list or create our own list.



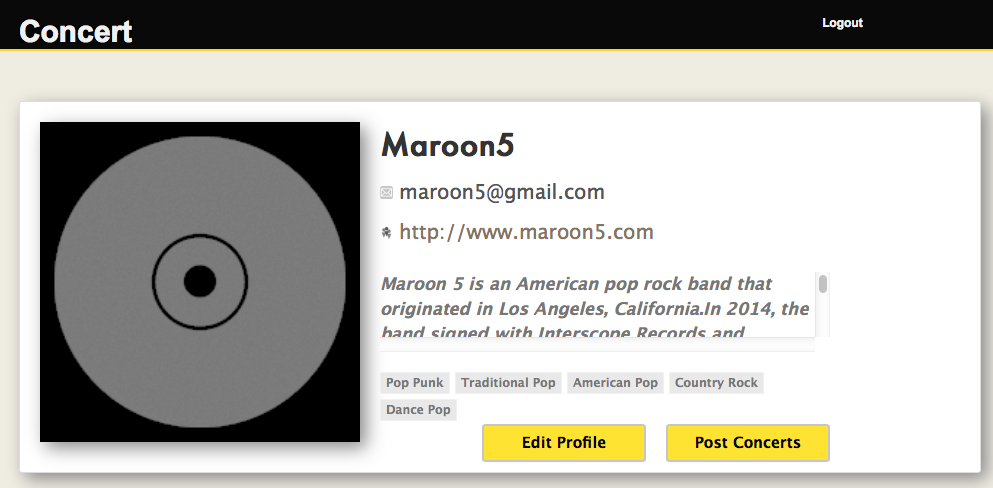
7. In connection page, we could see the follower and the followee



8. In my profile page, we could see & edit our profile and create list(lol), we could also see our membership length



9. Login as an artist. As an artist, you could only edit your profile or post an concert.



10.Post concert page.

