Project

CS 6083 Principles of Database Systems

Spring 2015

Pinterest – Final Report

Submitted by

Chandu Sridhar Gorthi
ID: N15512275
Email: csg350@nyu.edu

Sai Venkata Naga Manikanta Abhishek Talluri
ID: N13147813
Email: svt244@nyu.edu

References:

We have used a bootstrap theme called as metronic for CSS and other basic layout. The link for this theme is available in all the webpages.

We have referred to this project where the connection to the database is given.

https://github.com/joshcam/PHP-MySQLi-Database-Class

Description:

- We are designing a web based service similar to pinterest where the users can create a profile with their credentials and then create create boards for pinning pictures which they are interested in.
- Users can also follow boards by other users, which will help to populate their feed. Their feed contains the pictures that are pinned to the followed board.
- We are implementing a friendship mechanism where the users can become friends with other users and can restrict the access of commenting on their boards only to friends by specifying a board type as 'friend only' or 'public'.

Assumptions:

- Boards are considered as weak entities as they cannot exist without a user who creates them.

- Pictures can be existing even when they are not pinned onto any board,i.e when a user uploads an image and gets a URL, we might need in the later parts of the project to store this information even if the user hasn't pinned that image.

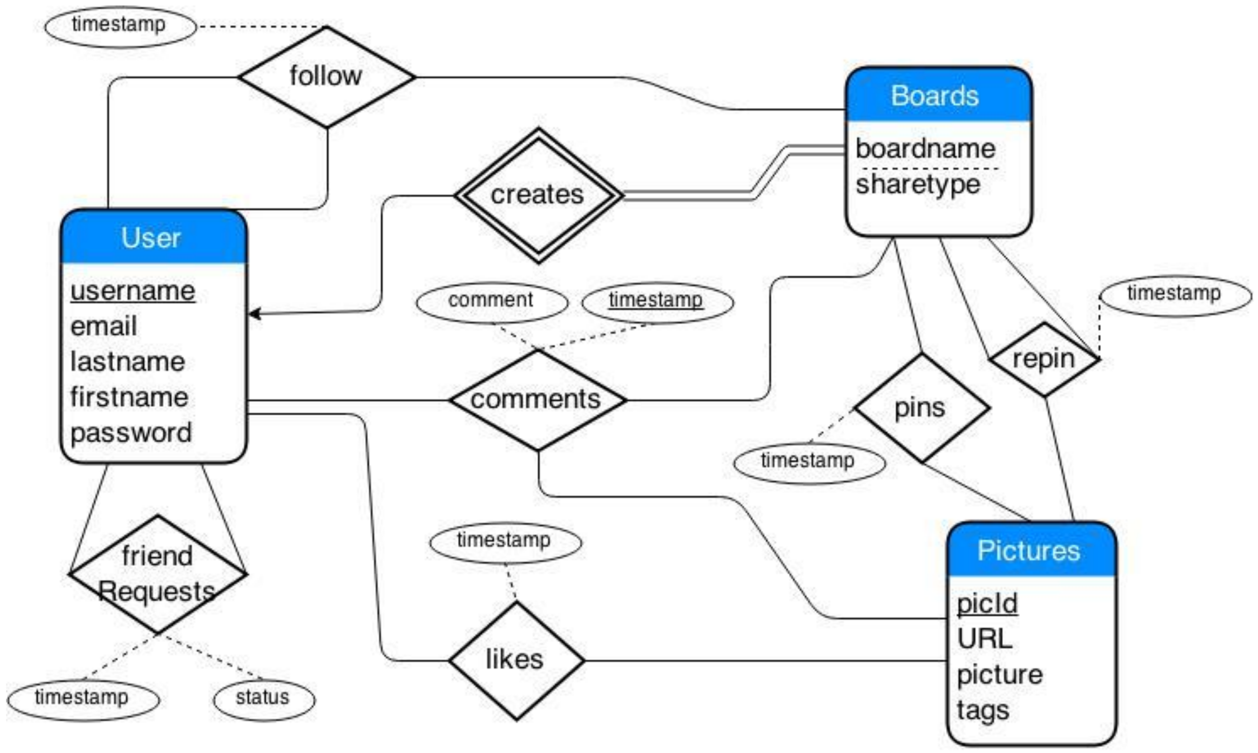
- All the repinned pictures are deleted in cascade when an original pinned picture has been deleted.

- Repinning a picture multiple times on the same board are considered as a single repin. If this has to be modelled as a different repin, we will be modifying this design in the second part by considering the timestamp as part of the primary key constraint in the repin table.

- Likes are implemented as a relation between the pictures and the users as the likes are counted upon the main pin instead of its individual repins.

- The follow option is modelled as a ternary relation as shown in the ER diagram and it is reduced to the relational schema as follow_stream(<u>username</u>, <u>username2</u>, <u>boardname2,</u> timestamp)

E-R Diagram:



Database Schema:

User (<u>username</u>, first name, last name, email (unique), password)

Pictures (<u>picId</u>, url (unique), picture, tags)

Pinboards (<u>boardname</u>, <u>username,</u> sharetype)
foreign key (username) references user (username)

Friends (<u>username, username2</u>, status (request sent, request accepted), timestamp)
foreign key (username) references user (username)
foreign key (username2) references user (username)

Pin (<u>picId</u>, <u>boardname</u>, <u>username,</u> timestamp)
foreign key (picId) references pictures (picId)
foreign key (boardname, username) references pinboards (boardname, username)

Follow_stream (<u>username</u>, <u>username2</u>, <u>boardname2,</u> timestamp)
foreign key (username) references user (username)
foreign key (username2, boardname2) references pinboards (username, boardname)

Re-pins (<u>username</u>, <u>boardname</u>, <u>picId</u>, <u>boardname2</u>, <u>username2,</u> timestamp)
foreign key (username, boardname) references pinboards (username, boardname)
foreign key (boardname2, username2) references pinboards (boardname, username)

foreign key (picId) references pictures (picId)

Likes (username, pictureid, timestamp)
foreign key (username) references user (username)
foreign key (picId) references pictures (picId)

Comments (username, username2, boardname2, pictureid, comment, timestamp)
foreign key (username2, boardname2) references pinboards (username, boardname)
foreign key (username) references user (username)
foreign key (picId) references pictures (picId)

Schema Definition:

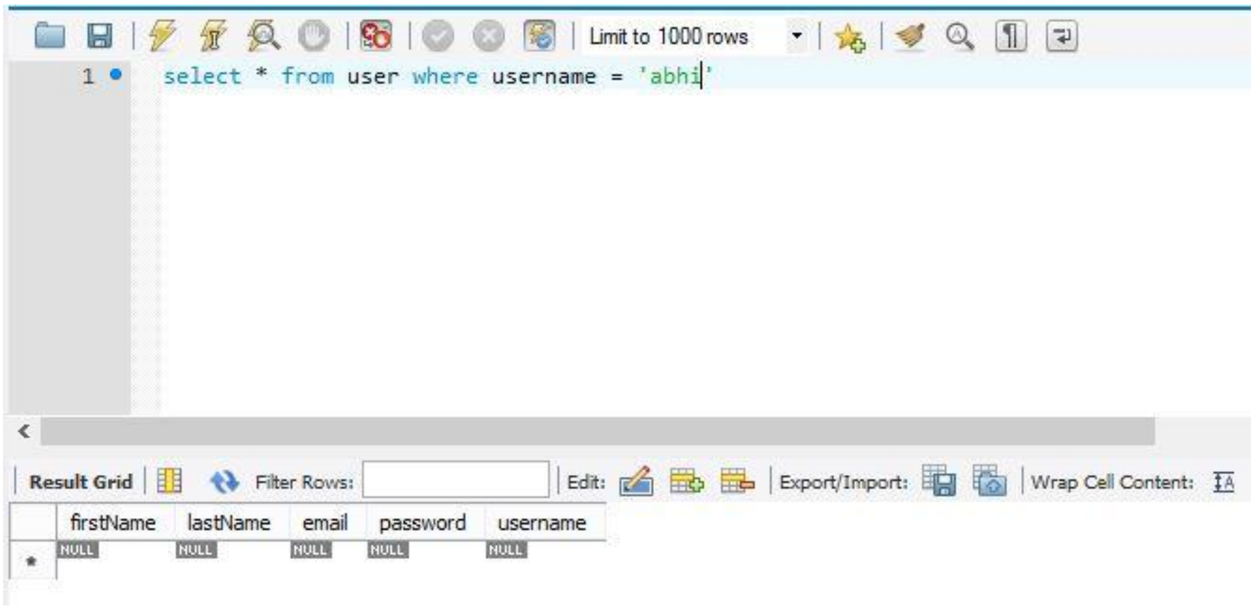
```
CREATE TABLE `user` (

  `firstName` varchar(20),

  `lastName` varchar(20),

  `email` varchar(50) NOT NULL,

  `password` varchar(20) NOT NULL,

  `username` varchar(25) NOT NULL,

  PRIMARY KEY (`username`),

  UNIQUE KEY `email_UNIQUE` (`email`)

);

CREATE TABLE `pinboards` (

  `boardname` varchar(15) NOT NULL,

  `username` varchar(25) NOT NULL,

  `sharetype` varchar(10) DEFAULT 'Public',

  PRIMARY KEY (`boardname`,`username`),

FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE
CASCADE

);
```

```
CREATE TABLE `pictures` (
  `picId` int(10) NOT NULL AUTO_INCREMENT,
  `url` varchar(200) NOT NULL,
  `picture` longblob NOT NULL,
  `tags` varchar(200) NOT NULL,
  `username` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`picId`)
);


CREATE TABLE `friends` (

  `username` varchar(25) NOT NULL,

  `username2` varchar(25) NOT NULL,

  `status` varchar(16) DEFAULT 'request sent',

  `timestamp` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

  PRIMARY KEY (`username`,`username2`),

  FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE
CASCADE,

  FOREIGN KEY (`username2`) REFERENCES `user` (`username`) ON DELETE CASCADE ON
UPDATE CASCADE

);

CREATE TABLE `pin` (

  `picId` int(10) NOT NULL,

  `boardname` varchar(15) NOT NULL,

  `username` varchar(25) NOT NULL,

  `timestamp` timestamp(6) NULL DEFAULT NULL,

  PRIMARY KEY (`picId`,`boardname`,`username`),

  KEY `boardDetails_idx` (`boardname`,`username`),

  FOREIGN KEY (`boardname`, `username`) REFERENCES `pinboards` (`boardname`, `username`)
ON DELETE CASCADE ON UPDATE CASCADE,

  FOREIGN KEY (`picId`) REFERENCES `pictures` (`picId`) ON DELETE CASCADE ON UPDATE
CASCADE
```

```
);

CREATE TABLE `streams` (
  `streamname` varchar(25) NOT NULL,
  `username` varchar(25) NOT NULL,
  PRIMARY KEY (`streamname`,`username`),
  KEY `username8` (`username`),
  CONSTRAINT `username8` FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON
DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE `follow_stream` (

  `username` varchar(25) NOT NULL,

  `username2` varchar(25) NOT NULL,

  `boardname2` varchar(15) NOT NULL,

  `timestamp` timestamp(6) NULL DEFAULT NULL,

  PRIMARY KEY (`username`,`username2`,`boardname2`),

  FOREIGN KEY (`username2`, `boardname2`) REFERENCES `pinboards` (`username`,
`boardname`) ON DELETE CASCADE ON UPDATE CASCADE,

  FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE
CASCADE

);

CREATE TABLE `re_pins` (

  `username` varchar(25) NOT NULL,

  `boardname` varchar(15) NOT NULL,

  `picId` int(10) NOT NULL,

  `boardname2` varchar(15) NOT NULL,

  `username2` varchar(25) NOT NULL,

  `timestamp` timestamp(6) NULL DEFAULT NULL,

  PRIMARY KEY (`username`,`boardname`,`picId`,`boardname2`,`username2`),

  FOREIGN KEY (`username`, `boardname`) REFERENCES `pinboards` (`username`, `boardname`)
ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`picId`, `boardname2`, `username2`) REFERENCES `pin` (`picId`, `boardname`,
`username`) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE `likes` (

  `username` varchar(25) NOT NULL,

  `picId` int(10) NOT NULL,

  `timestamp` timestamp(6) NULL DEFAULT NULL,

  PRIMARY KEY (`username`,`picId`),

  FOREIGN KEY (`picId`) REFERENCES `pictures` (`picId`) ON DELETE CASCADE ON UPDATE
CASCADE,

  FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE
CASCADE

);

CREATE TABLE `comments` (

  `username` varchar(25) NOT NULL,

  `username2` varchar(25) NOT NULL,

  `boardname2` varchar(15) NOT NULL,

  `picId` int(10) NOT NULL,

  `comment` varchar(200) DEFAULT NULL,

  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

  PRIMARY KEY (`username`,`username2`,`boardname2`,`picId`,`timestamp`),

  FOREIGN KEY (`picId`, `boardname2`, `username2`) REFERENCES `pin` (`picId`, `boardname`,
`username`) ON DELETE CASCADE ON UPDATE CASCADE,

  FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE
CASCADE

);
```

Sample Queries:

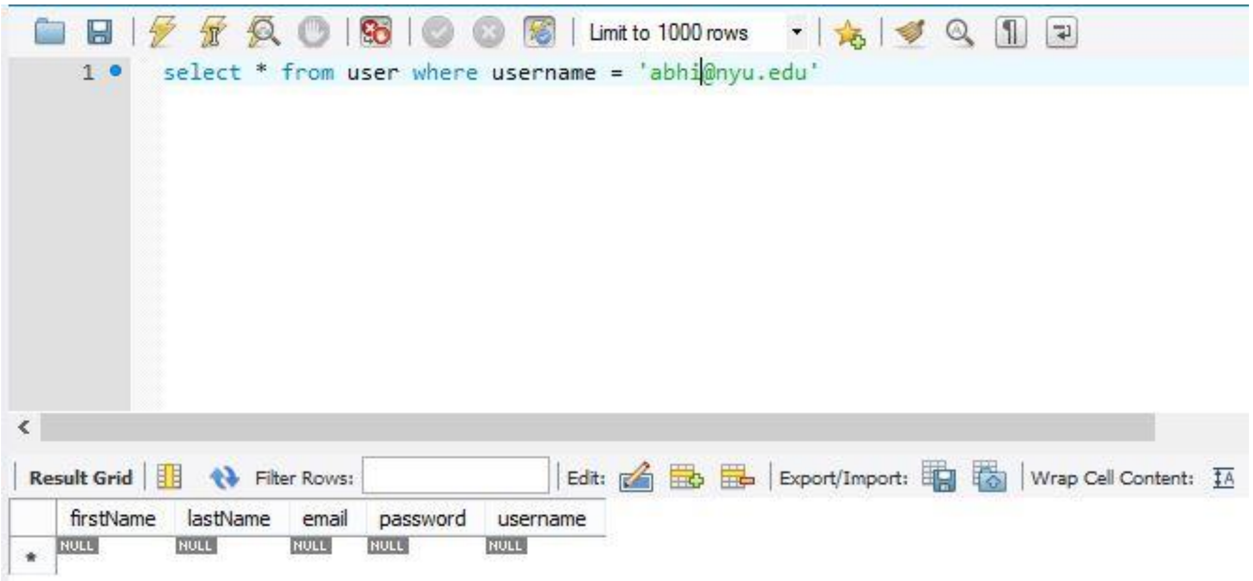
1)
sign up:
select * from user where username = 'abhi'



```
select * from user where username = 'abhi'
```

	firstName	lastName	email	password	username
*	NULL	NULL	NULL	NULL	NULL

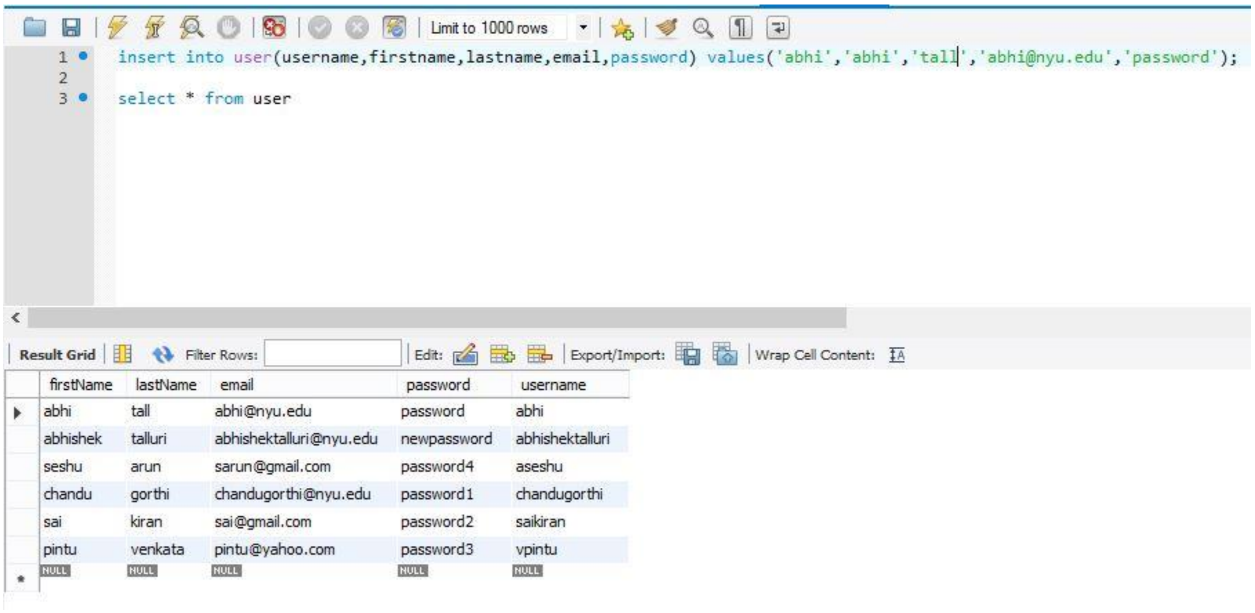
select * from user where email = 'abhi@nyu.edu'



```
select * from user where username = 'abhi@nyu.edu'
```

	firstName	lastName	email	password	username
*	NULL	NULL	NULL	NULL	NULL

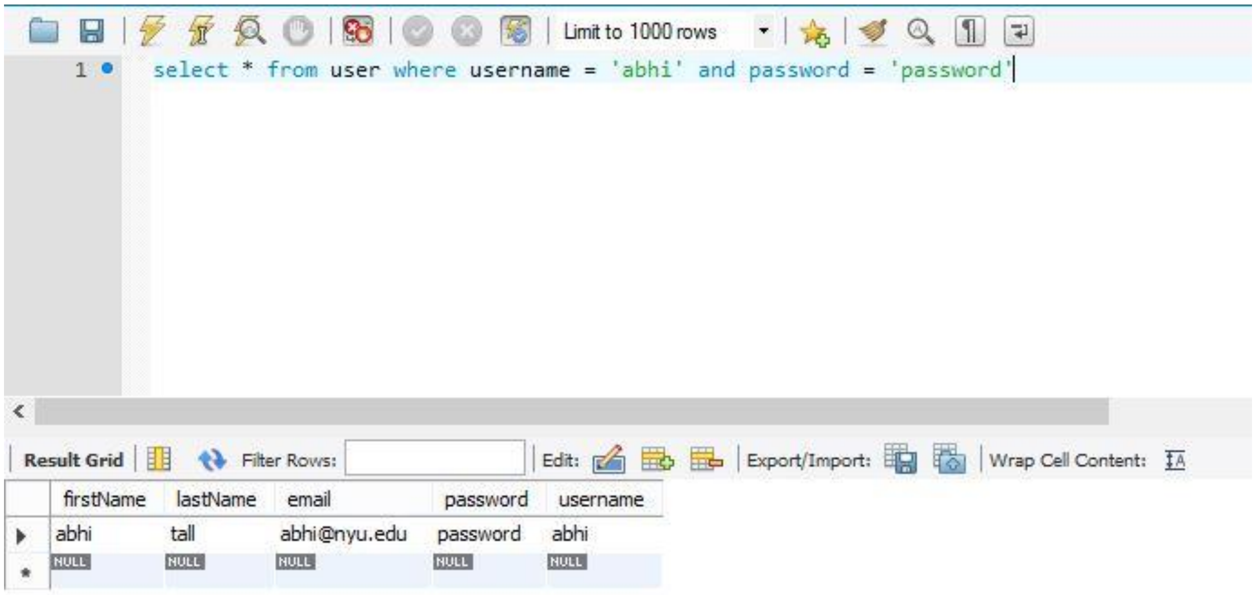
Both these queries should return null unless which the user can sign up with the given details, or the
user have to provide different username or email address.
then we can insert into the user's database
insert into user(username,firstname,lastname,email,password)
values('abhi','abhi','tall','abhi@nyu.edu','password');



```
insert into user(username,firstname,lastname,email,password) values('abhi','abhi','tall','abhi@nyu.edu','password');

select * from user
```

firstName	lastName	email	password	username
abhi	tall	abhi@nyu.edu	password	abhi
abhishek	talluri	abhishektalluri@nyu.edu	newpassword	abhishektalluri
seshu	arun	sarun@gmail.com	password4	aseshu
chandu	gorthi	chandugorthi@nyu.edu	password1	chandugorthi
sai	kiran	sai@gmail.com	password2	saikiran
pintu	venkata	pintu@yahoo.com	password3	vpintu
NULL	NULL	NULL	NULL	NULL

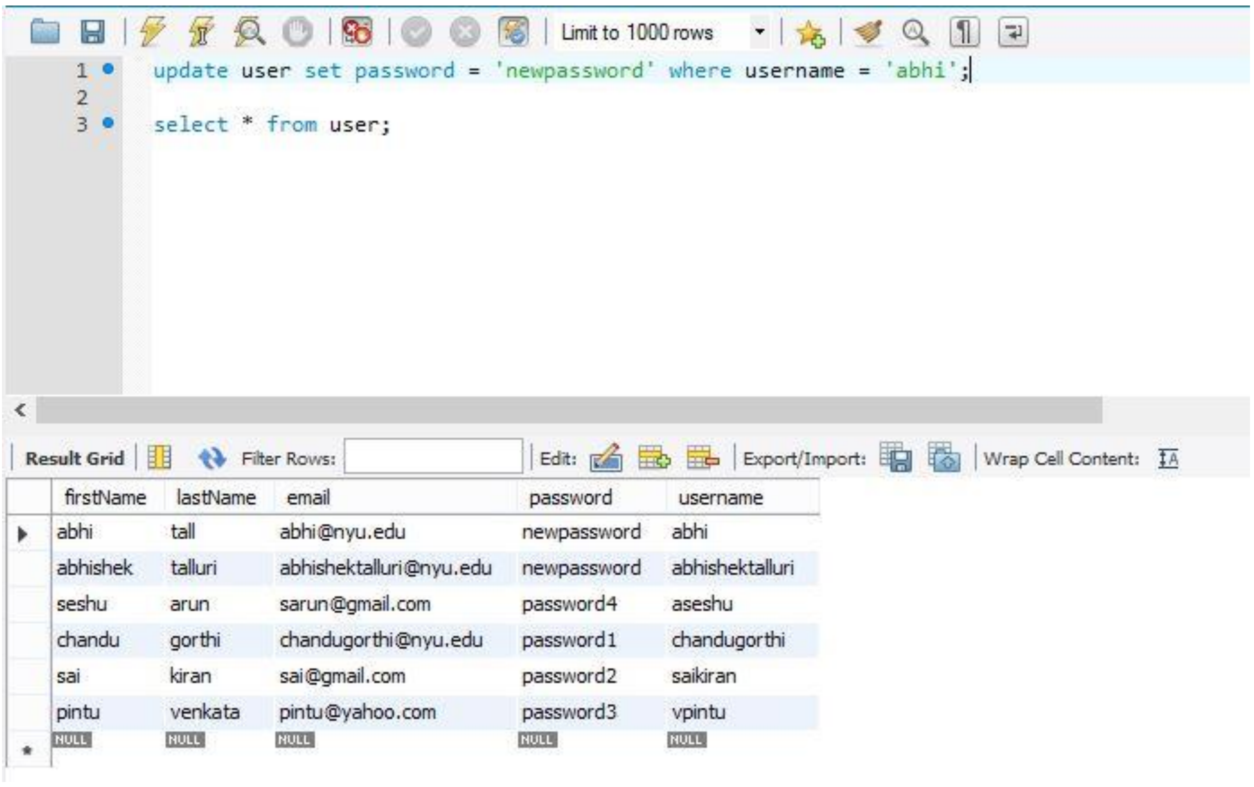
login: select * from user where username = 'abhi' and password = 'password'
If the user with username = abhi and password = password tries to login it has to match one of the existing entries in the database.



```
select * from user where username = 'abhi' and password = 'password'
```

firstName	lastName	email	password	username
abhi	tall	abhi@nyu.edu	password	abhi
NULL	NULL	NULL	NULL	NULL

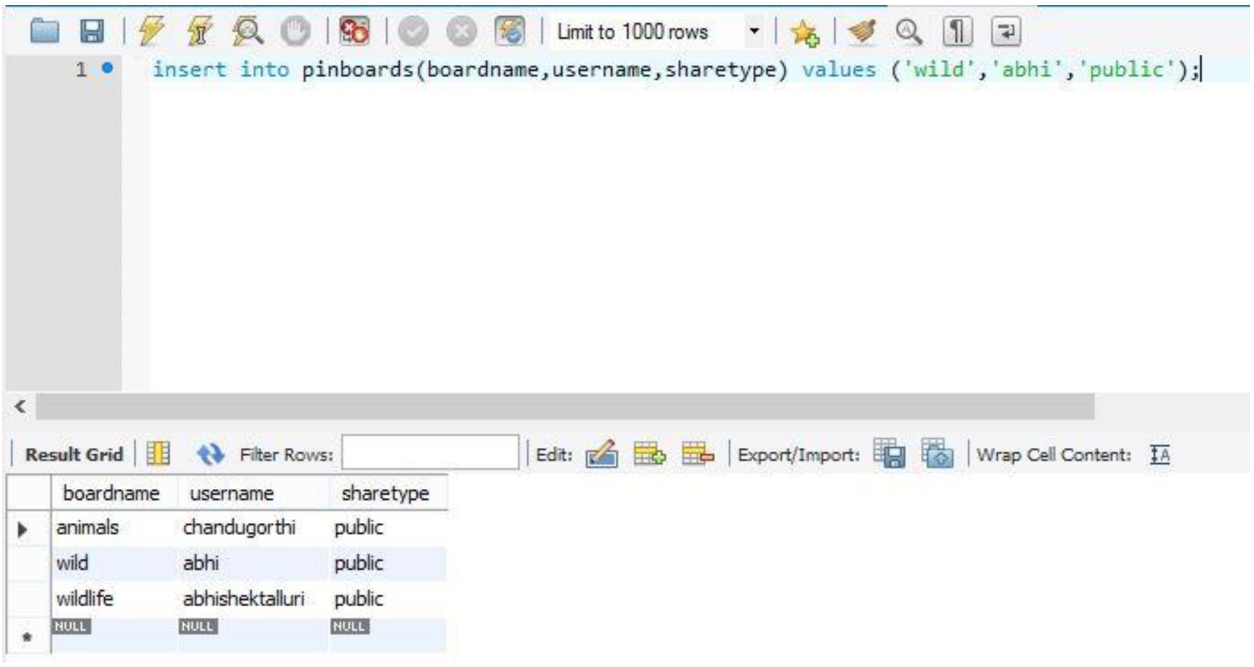
edit their profile:
update user set password = 'newpassword' where username = 'abhi'



```
update user set password = 'newpassword' where username = 'abhi';

select * from user;
```

firstName	lastName	email	password	username
abhi	tall	abhi@nyu.edu	newpassword	abhi
abhishek	talluri	abhishektalluri@nyu.edu	newpassword	abhishektalluri
seshu	arun	sarun@gmail.com	password4	aseshu
chandu	gorthi	chandugorthi@nyu.edu	password1	chandugorthi
sai	kiran	sai@gmail.com	password2	saikiran
pintu	venkata	pintu@yahoo.com	password3	vpintu
NULL	NULL	NULL	NULL	NULL

when a user is logged into his account, and when he tries to create a pinboard,
insert into pinboards(boardname,username,sharetype) values ('wild','abhi','public');

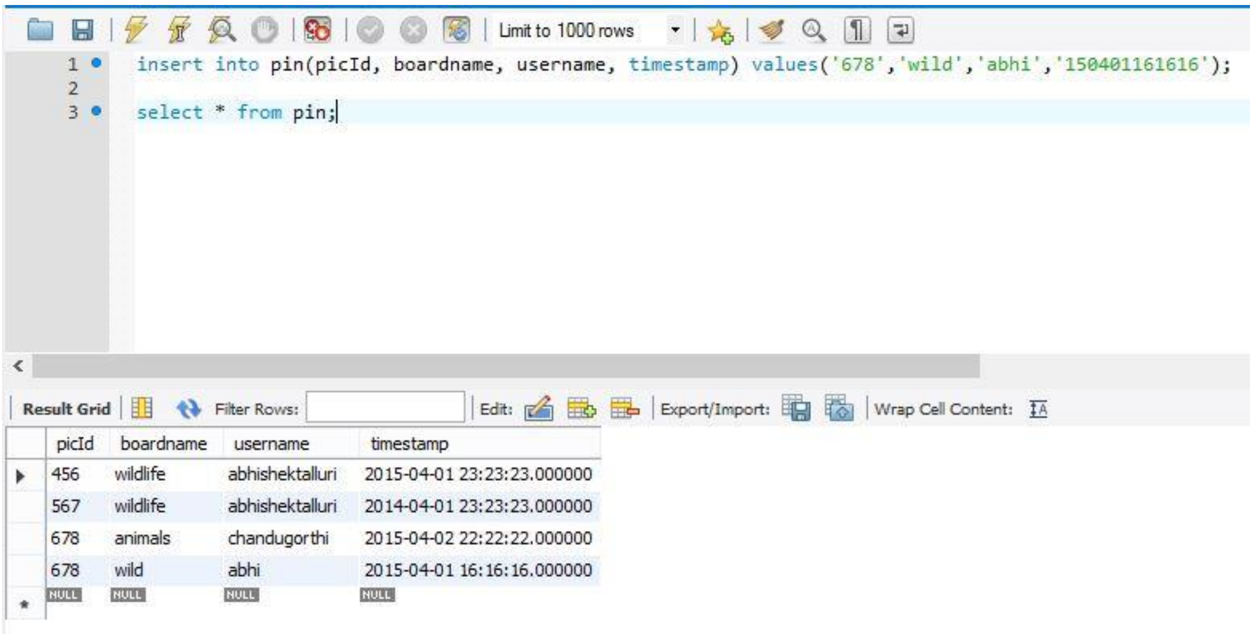


```
insert into pinboards(boardname,username,sharetype) values ('wild','abhi','public');
```

	boardname	username	sharetype
▶	animals	chandugorthi	public
	wild	abhi	public
	wildlife	abhishektalluri	public
*	NULL	NULL	NULL

To pin a picture to the wildlife board, we assume that there is an entry that has been inserted in the picture table with values:
('678','www.wildlife.com/lion.jpg',null,'wildlife,lion,zoo,forest')

insert into pin(picId, boardname, username, timestamp)
values('678','wild','abhi','150401161616')

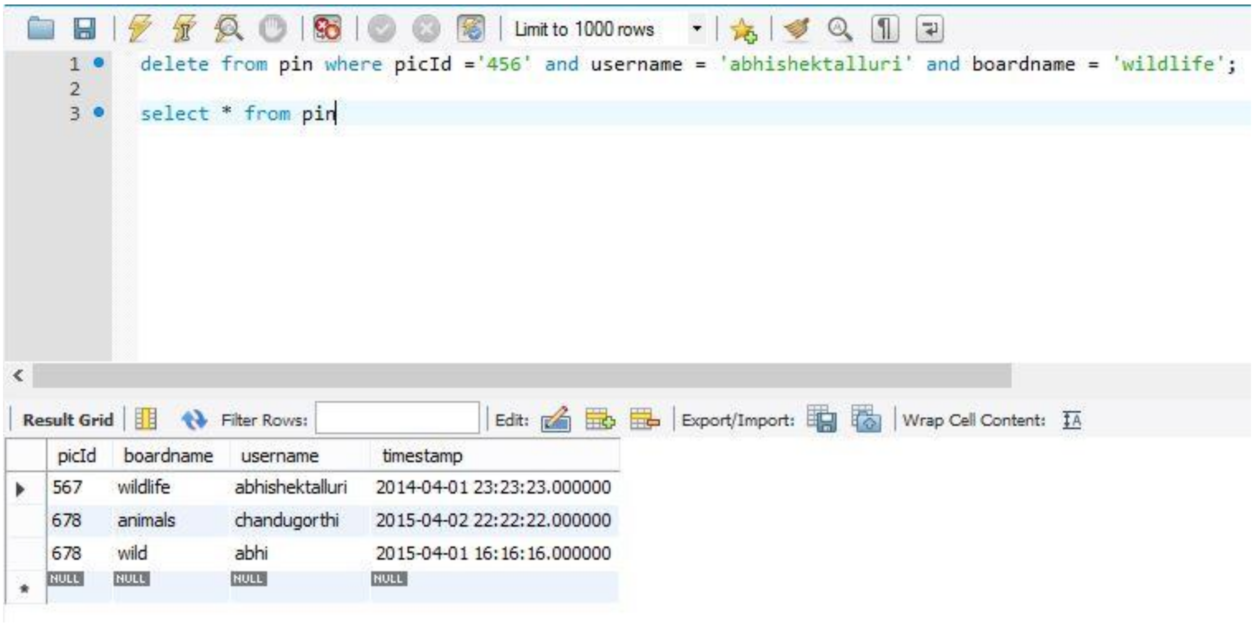


```
insert into pin(picId, boardname, username, timestamp) values('678','wild','abhi','150401161616');

select * from pin;
```

	picId	boardname	username	timestamp
▶	456	wildlife	abhishektalluri	2015-04-01 23:23:23.000000
	567	wildlife	abhishektalluri	2014-04-01 23:23:23.000000
	678	animals	chandugorthi	2015-04-02 22:22:22.000000
	678	wild	abhi	2015-04-01 16:16:16.000000
*	NULL	NULL	NULL	NULL

When a user deletes a pinned picture from one of his boards, we remove it from the pins table.

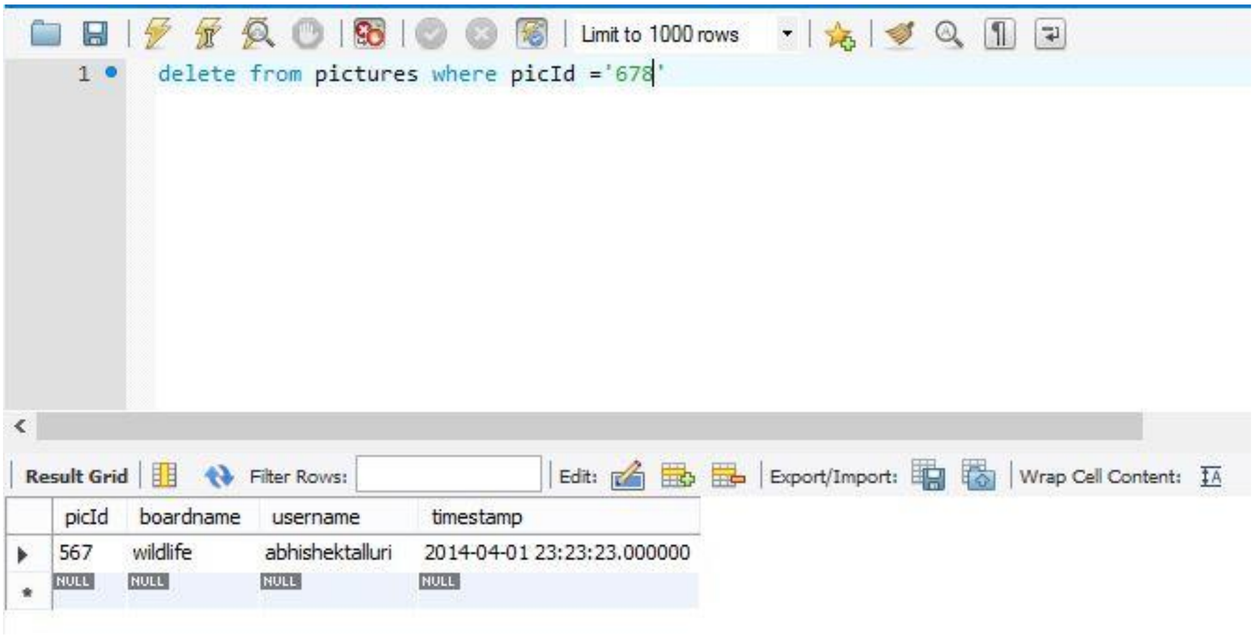
delete from pin where picId ='456' and username = 'abhishektalluri' and boardname = 'wildlife';



```
delete from pin where picId ='456' and username = 'abhishektalluri' and boardname = 'wildlife';

select * from pin
```

picId	boardname	username	timestamp
567	wildlife	abhishektalluri	2014-04-01 23:23:23.000000
678	animals	chandugorthi	2015-04-02 22:22:22.000000
678	wild	abhi	2015-04-01 16:16:16.000000
NULL	NULL	NULL	NULL

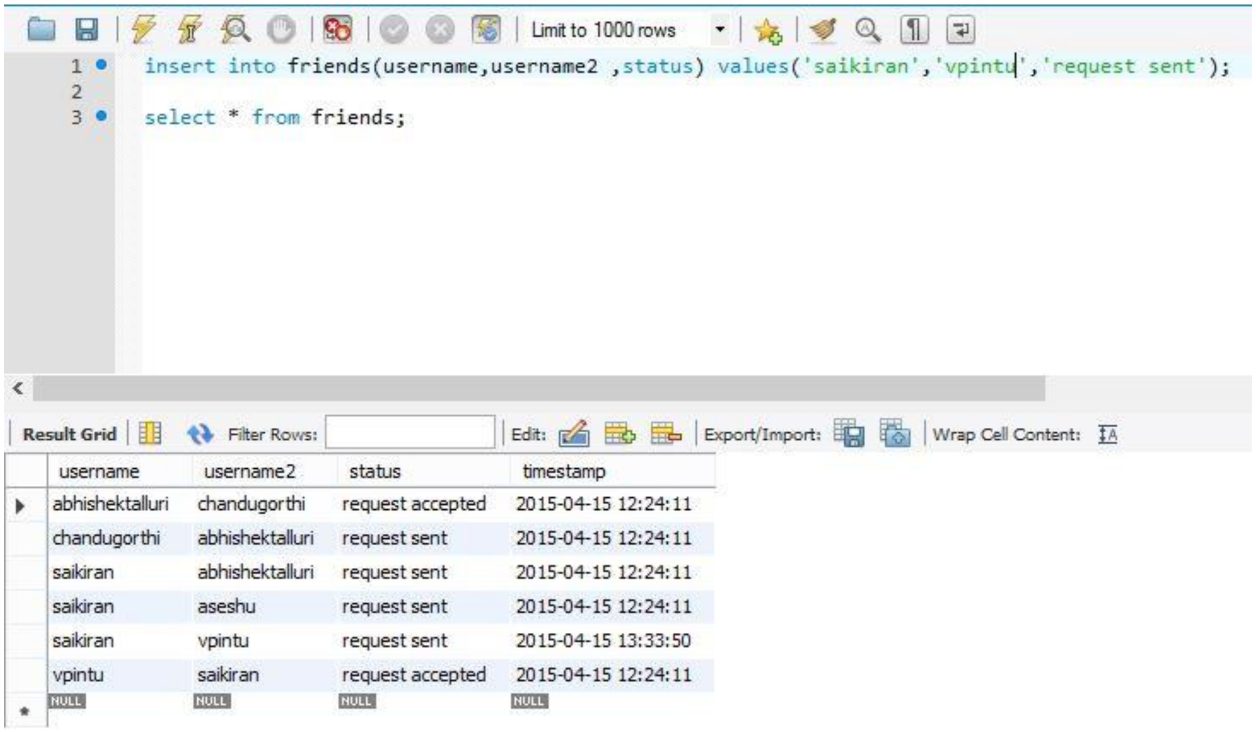
When a picture is deleted from pictures table all its occurrences are deleted from pin table
delete from pictures where picId ='678'



```
delete from pictures where picId ='678'
```

picId	boardname	username	timestamp
567	wildlife	abhishektalluri	2014-04-01 23:23:23.000000
NULL	NULL	NULL	NULL

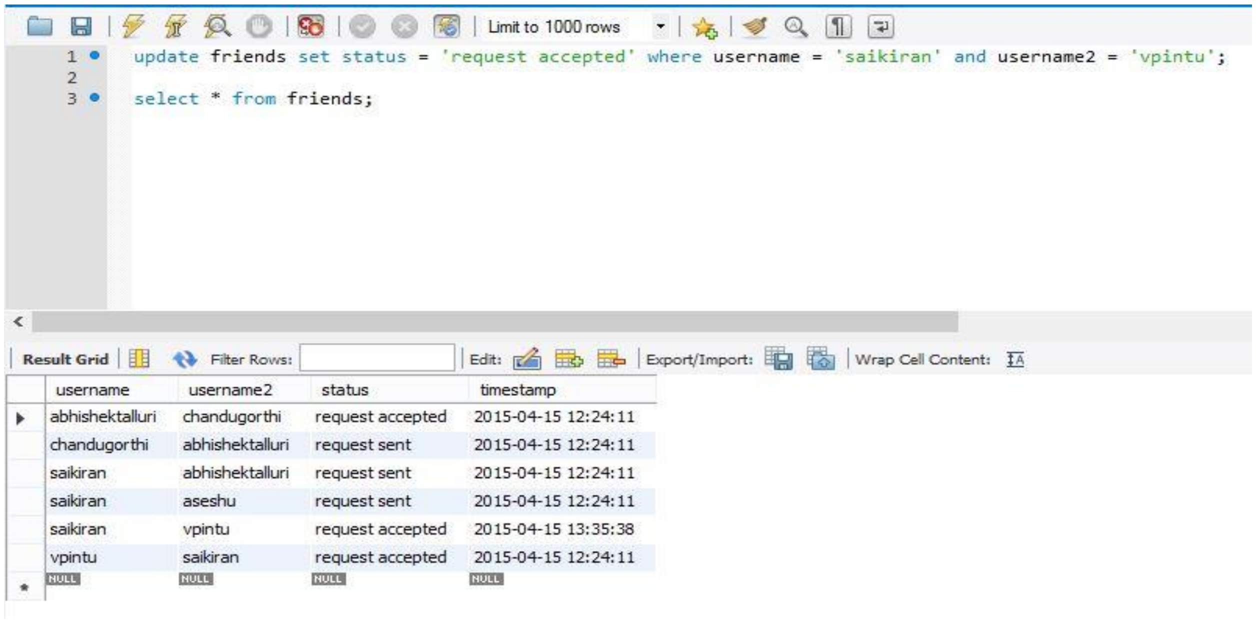
2) when a user sends a friend request to other user we add an entry into the friends table with the field of status as 'request sent'
insert into friends(username,username2 ,status) values('saikiran','vpintu','request sent');



```
insert into friends(username,username2 ,status) values('saikiran','vpintu','request sent');

select * from friends;
```

username	username2	status	timestamp
abhishektalluri	chandugorthi	request accepted	2015-04-15 12:24:11
chandugorthi	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	aseshu	request sent	2015-04-15 12:24:11
saikiran	vpintu	request sent	2015-04-15 13:33:50
vpintu	saikiran	request accepted	2015-04-15 12:24:11
NULL	NULL	NULL	NULL

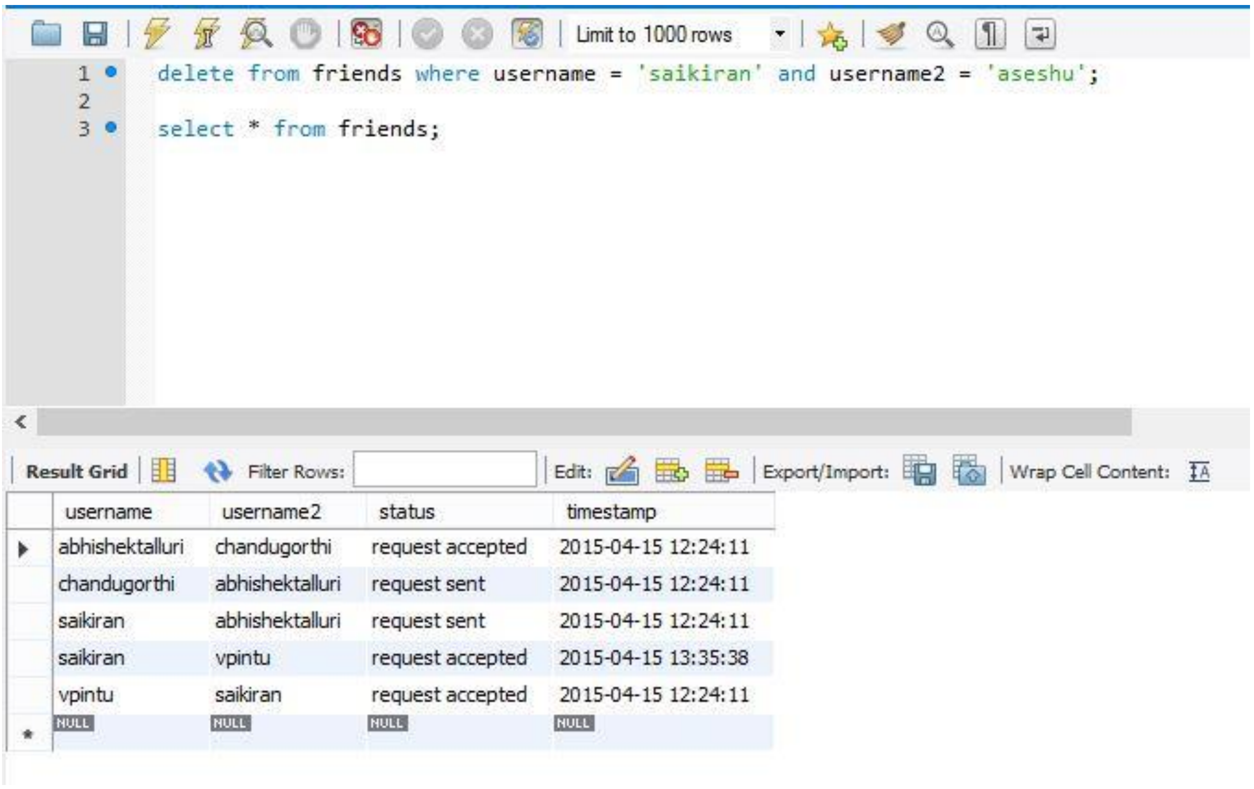
when the user accepts the request and we update the status field in the friends table status field to 'request accepted'
update friends set status = 'request accepted' where username = 'saikiran' and username2 = 'vpintu';



```
update friends set status = 'request accepted' where username = 'saikiran' and username2 = 'vpintu';

select * from friends;
```

username	username2	status	timestamp
abhishektalluri	chandugorthi	request accepted	2015-04-15 12:24:11
chandugorthi	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	aseshu	request sent	2015-04-15 12:24:11
saikiran	vpintu	request accepted	2015-04-15 13:35:38
vpintu	saikiran	request accepted	2015-04-15 12:24:11
NULL	NULL	NULL	NULL

when the user declines a request from another user, then we delete that entry from the friends table.
delete from friends where username = 'saikiran' and username2 = 'aseshu';

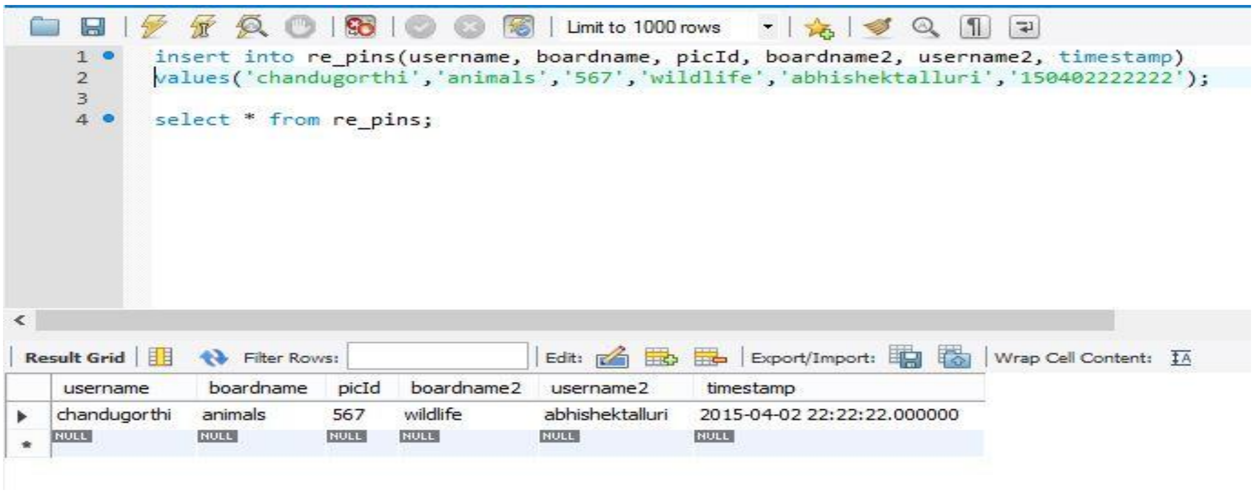


```
delete from friends where username = 'saikiran' and username2 = 'aseshu';

select * from friends;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

username	username2	status	timestamp
abhishektalluri	chandugorthi	request accepted	2015-04-15 12:24:11
chandugorthi	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	vpintu	request accepted	2015-04-15 13:35:38
vpintu	saikiran	request accepted	2015-04-15 12:24:11
NULL	NULL	NULL	NULL

3) when a user named 'chandugorthi' tries to repin a picture that is found on abhishektalluri's pinboard 'wildlife' into his board named 'animals', then we add an entry into the repin table as
insert into re_pins(username, boardname, picId, boardname2, username2, timestamp) values('chandugorthi','animals','567','wildlife','abhishektalluri','150402222222');

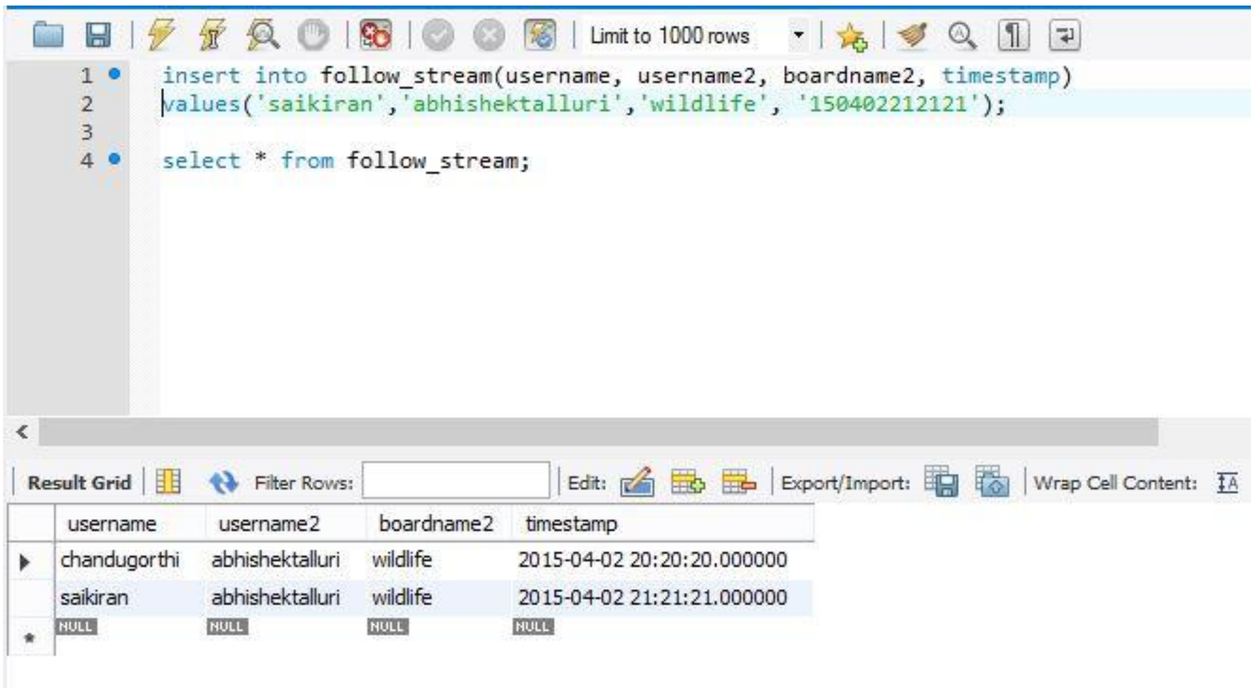


```
insert into re_pins(username, boardname, picId, boardname2, username2, timestamp)
values('chandugorthi','animals','567','wildlife','abhishektalluri','150402222222');

select * from re_pins;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

username	boardname	picId	boardname2	username2	timestamp
chandugorthi	animals	567	wildlife	abhishektalluri	2015-04-02 22:22:22.000000
NULL	NULL	NULL	NULL	NULL	NULL

4)
To create a follow stream, the user has to follow a board on another user.
If a user named 'chandugorthi' follows a 'wildlife' board created by user 'abhishektalluri', we add the following entry into the table follow_streams
insert into follow_stream(username, username2, boardname2, timestamp)
values('saikiran','abhishektalluri','wildlife', '150402212121');

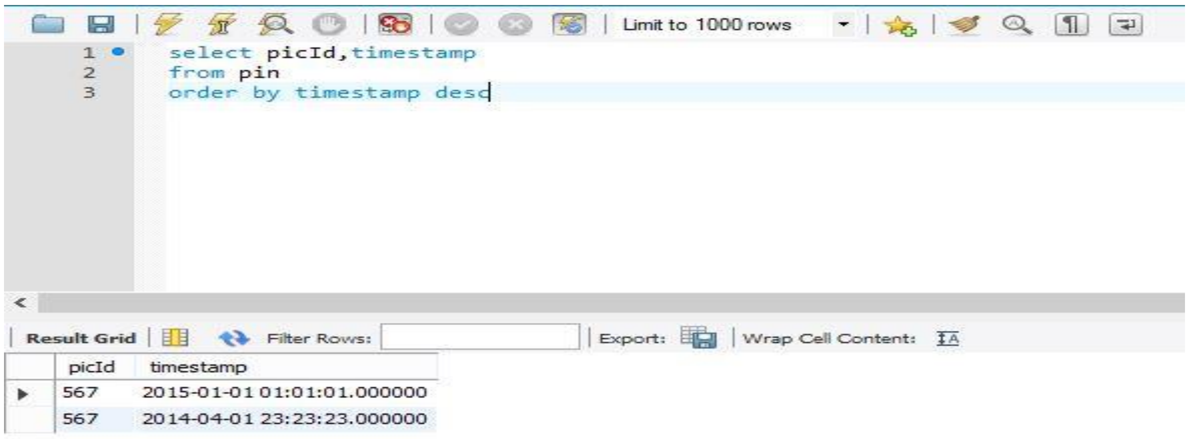


```
insert into follow_stream(username, username2, boardname2, timestamp)
values('saikiran','abhishektalluri','wildlife', '150402212121');

select * from follow_stream;
```

username	username2	boardname2	timestamp
chandugorthi	abhishektalluri	wildlife	2015-04-02 20:20:20.000000
saikiran	abhishektalluri	wildlife	2015-04-02 21:21:21.000000
NULL	NULL	NULL	NULL

To display all the pictures in a follow stream in reverse chronological order, we can use the boardname and username to get the pictures pinned to that board.To display the pinned pictures to a particular board, we need to get the URL's for that pictures.

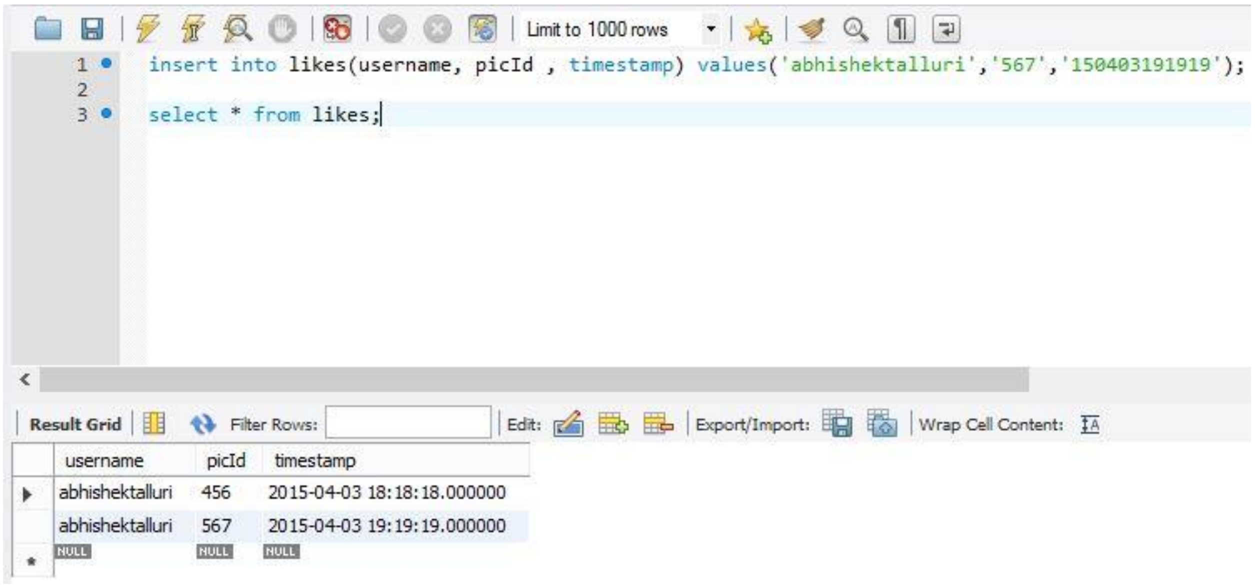
select picId,timestamp
from pin
order by timestamp desc



```
select picId,timestamp
from pin
order by timestamp desc
```

picId	timestamp
567	2015-01-01 01:01:01.000000
567	2014-04-01 23:23:23.000000

5)
when a user likes a picture, we insert a row into the likes table,
insert into likes(username, picId , timestamp) values('abhishektalluri','567','150403191919');



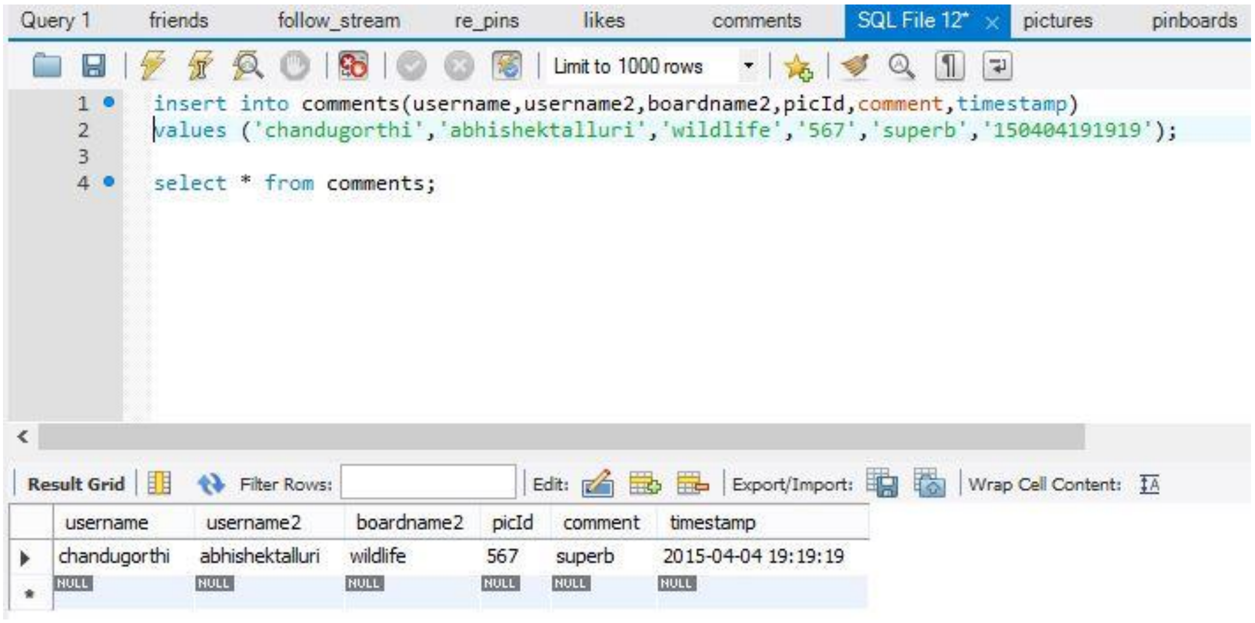
Limit to 1000 rows

```
insert into likes(username, picId , timestamp) values('abhishektalluri','567','150403191919');

select * from likes;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

username	picId	timestamp
abhishektalluri	456	2015-04-03 18:18:18.000000
abhishektalluri	567	2015-04-03 19:19:19.000000
NULL	NULL	NULL

when there are no constraints for a user to comment on a picture on a specific board, the user can comment on that picture and we insert a row into the comments table.
when a user comments on a picture, we insert a row into the comments table
insert into comments(username,username2,boardname2,picId,comment,timestamp) values ('chandugorthi','abhishektalluri','wildlife','456','superb','150404191919')



Query 1 | friends | follow_stream | re_pins | likes | comments | SQL File 12* | pictures | pinboards

Limit to 1000 rows

```
insert into comments(username,username2,boardname2,picId,comment,timestamp)
values ('chandugorthi','abhishektalluri','wildlife','567','superb','150404191919');

select * from comments;
```

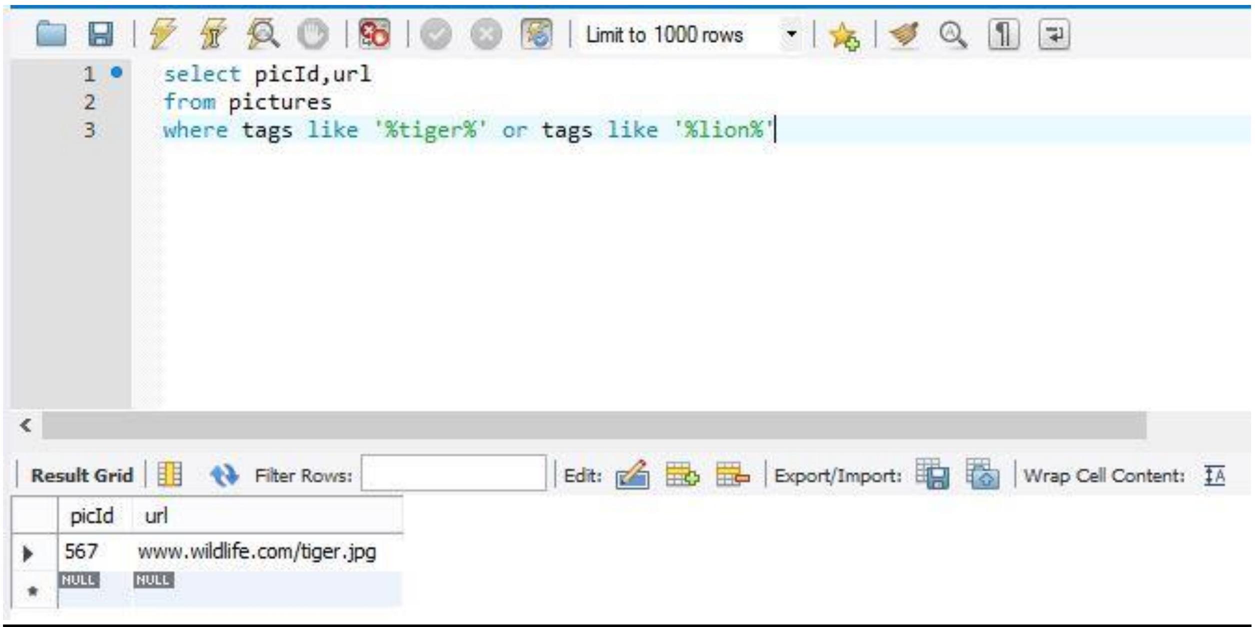
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

username	username2	boardname2	picId	comment	timestamp
chandugorthi	abhishektalluri	wildlife	567	superb	2015-04-04 19:19:19
NULL	NULL	NULL	NULL	NULL	NULL

If there are constraints for a board as to only the friends can pass in a comment, then we have this stored procedure where we first check the friendship status between the users and then add the comment to the comments table if they are friends.

The stored procedure for this insert is given in the stored procedures section in the latter part of this document.

6)
select picId,url
from pictures
where tags like '%keyword1% or tags like %keyword2%



```
select picId,url
from pictures
where tags like '%tiger%' or tags like '%lion%'
```

picId	url
567	www.wildlife.com/tiger.jpg
NULL	NULL

TRIGGERS:

1. CREATE DEFINER = CURRENT_USER TRIGGER `proj_pinterest`.`friends_BEFORE_INSERT` BEFORE INSERT ON `friends` FOR EACH ROW

```
   Begin
                    if new.`status` NOT IN ('request sent' , 'request accepted') then
                              signal sqlstate '44000'
         set message_text = 'status column is not valid';
                    end if ;
   end;
USE `proj_pinterest`;


DELIMITER $$
```

2. DROP TRIGGER IF EXISTS proj_pinterest.friends_BEFORE_INSERT$$

```
USE `proj_pinterest`$$
CREATE DEFINER = CURRENT_USER TRIGGER `proj_pinterest`.`friends_BEFORE_INSERT` BEFORE
INSERT ON `friends` FOR EACH ROW
   Begin
                    if new.`status` NOT IN ('request sent' , 'request accepted') then
                              signal sqlstate '44000'
         set message_text = 'status column is not valid';
                    end if ;
   end;$$
DELIMITER ;


DELIMITER $$
```

<u>Stored Procedures:</u>

1. create procedure **signup**(in uname varchar(25),in fname varchar(20), in lname varchar(20), in emailid varchar(50),in paswd varchar(20),inout flag int)

```
begin

        declare count int default 0;

        select count(*) into count from user where user.username= uname;

        if count<1 then

                insert into user values(fname,lname,emailid,paswd,uname);

        else set flag=1;

        end if;

end $$

DELIMITER ;
```

2. DELIMITER $$

```
create procedure checkFriend(in `uname` VARCHAR(25), in `oname` VARCHAR(25), inout status
int)
BEGIN
DECLARE status int default 0;
select count(*) into status from friends where (username='uname' and
username2='ouname') or (username='oname' and username2='uname');
END$$
DELIMITER ;
```

3. DELIMITER$$

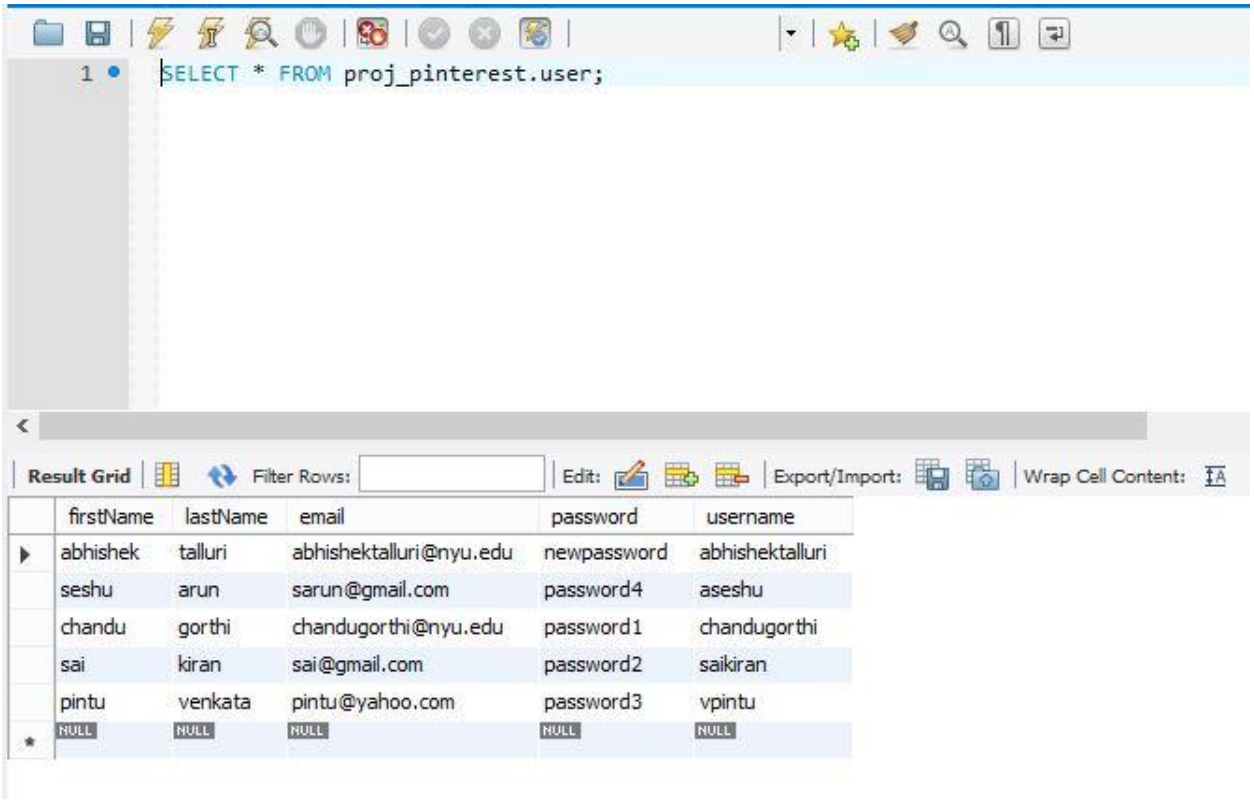
```
create procedure checkShare(in `uname`, in `bname`, inout `share`)
BEGIN
DECLARE count int default 0;
Select count(*) into count from pinboard where username = @uname and boardname =
@bname and sharetype = ‘public’
If count then
        Set @share = 1;
End if;
END$$
DELIMITER;
```

4. DELIMITER$$

```
create procedure comment(in `uname`, in `oname`, in `bname`, in `pic`, in `com`)
BEGIN
DECLARE status1 int default 0;
DECLARE status2 int default 0;
call checkFriend(@uname,@oname,status1);
call checkShare(@oname,@bname,@status2);
If @status1 AND @status2 then
        insert into comments(username,username2,boardname2,picId,comment,timestamp)
values (@uname,@oname,@bname,@pic,@com)
END$$
DELIMITER;
```

<u>Test Data for some important tables:</u>

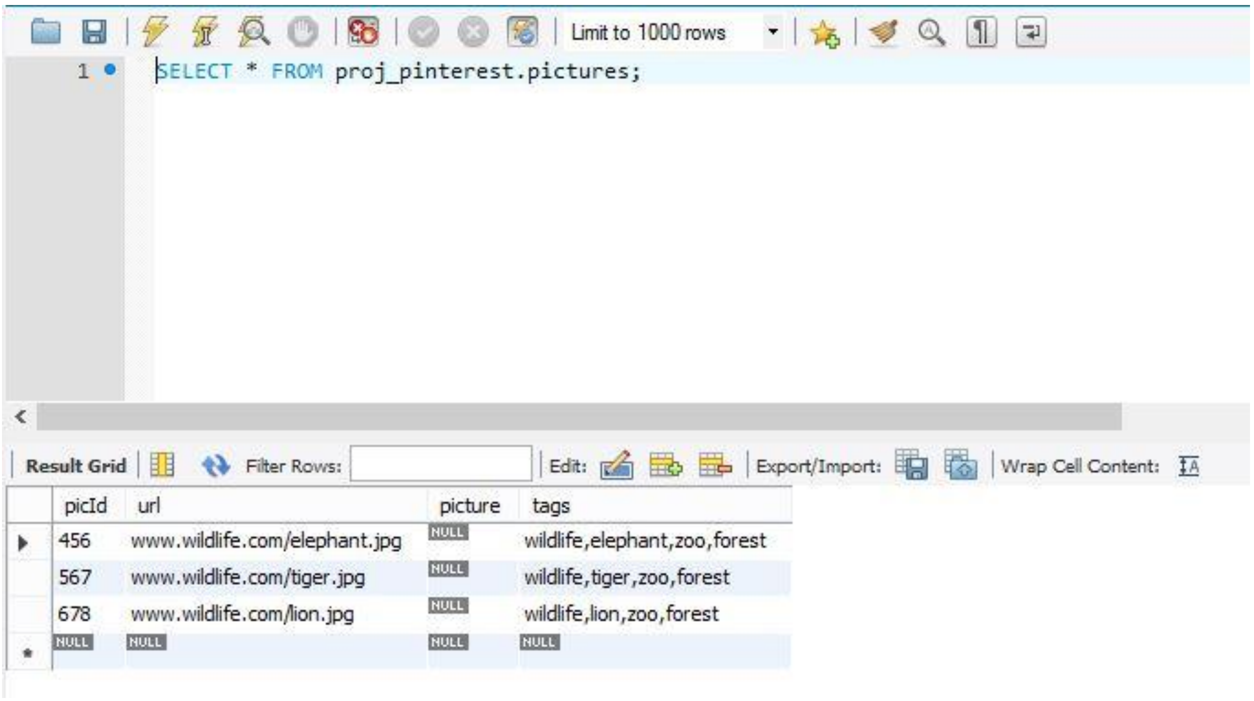
1. User:



```
SELECT * FROM proj_pinterest.user;
```

firstName	lastName	email	password	username
abhishek	talluri	abhishektalluri@nyu.edu	newpassword	abhishektalluri
seshu	arun	sarun@gmail.com	password4	aseshu
chandu	gorthi	chandugorthi@nyu.edu	password1	chandugorthi
sai	kiran	sai@gmail.com	password2	saikiran
pintu	venkata	pintu@yahoo.com	password3	vpintu
NULL	NULL	NULL	NULL	NULL

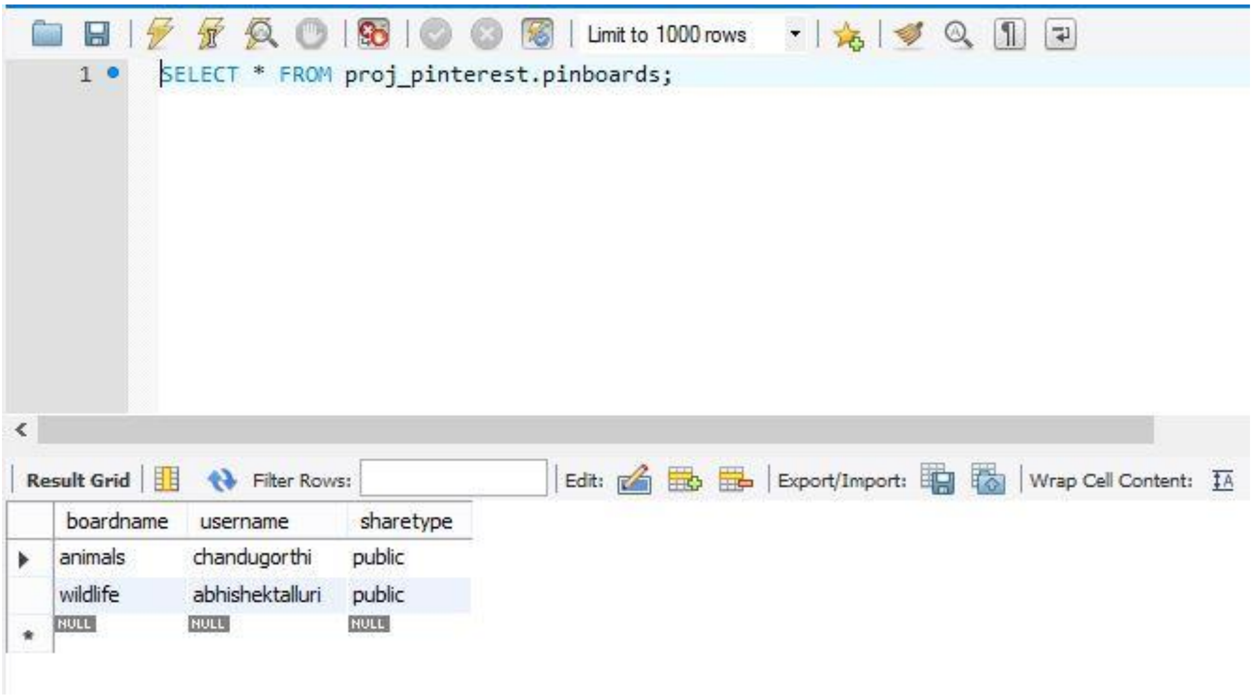
2. Pictures:



```
SELECT * FROM proj_pinterest.pictures;
```

picId	url	picture	tags
456	www.wildlife.com/elephant.jpg	NULL	wildlife,elephant,zoo,forest
567	www.wildlife.com/tiger.jpg	NULL	wildlife,tiger,zoo,forest
678	www.wildlife.com/lion.jpg	NULL	wildlife,lion,zoo,forest
NULL	NULL	NULL	NULL

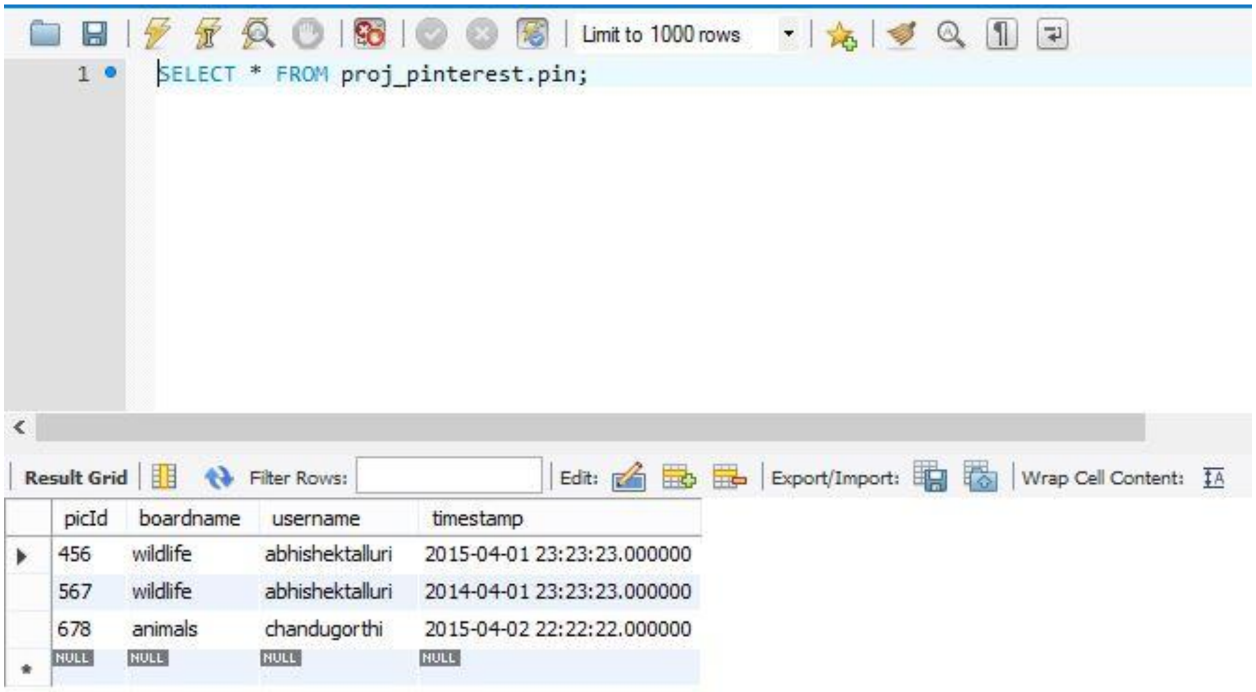
3. Pinboard:



```
SELECT * FROM proj_pinterest.pinboards;
```

boardname	username	sharetype
animals	chandugorthi	public
wildlife	abhishektalluri	public
NULL	NULL	NULL

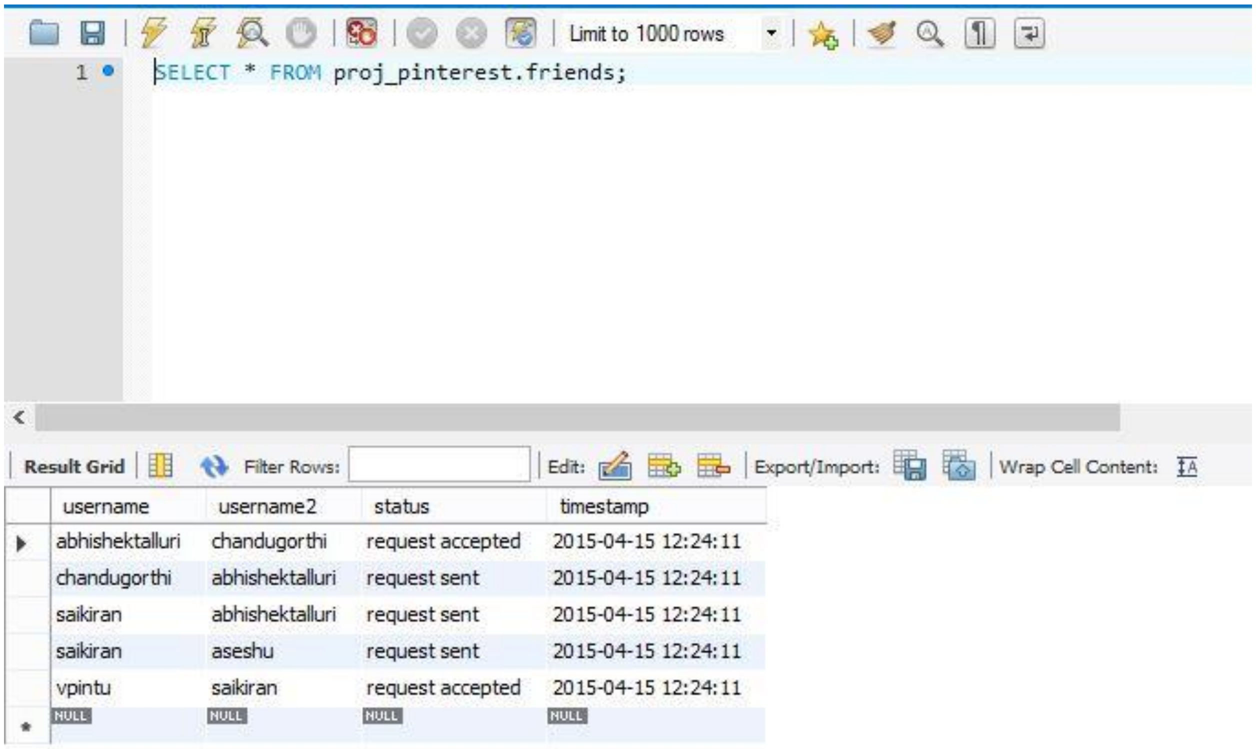
4. Pin



```
SELECT * FROM proj_pinterest.pin;
```

picId	boardname	username	timestamp
456	wildlife	abhishektalluri	2015-04-01 23:23:23.000000
567	wildlife	abhishektalluri	2014-04-01 23:23:23.000000
678	animals	chandugorthi	2015-04-02 22:22:22.000000
NULL	NULL	NULL	NULL

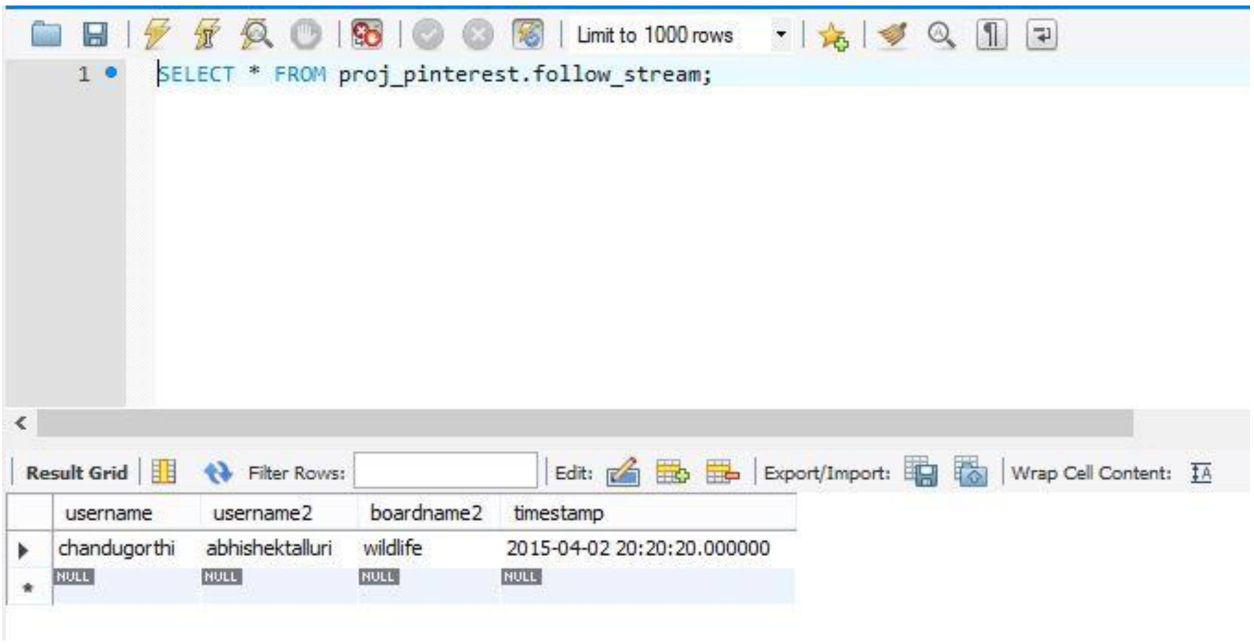
5. Friends:



```
SELECT * FROM proj_pinterest.friends;
```

username	username2	status	timestamp
abhishektalluri	chandugorthi	request accepted	2015-04-15 12:24:11
chandugorthi	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	abhishektalluri	request sent	2015-04-15 12:24:11
saikiran	aseshu	request sent	2015-04-15 12:24:11
vpintu	saikiran	request accepted	2015-04-15 12:24:11
NULL	NULL	NULL	NULL

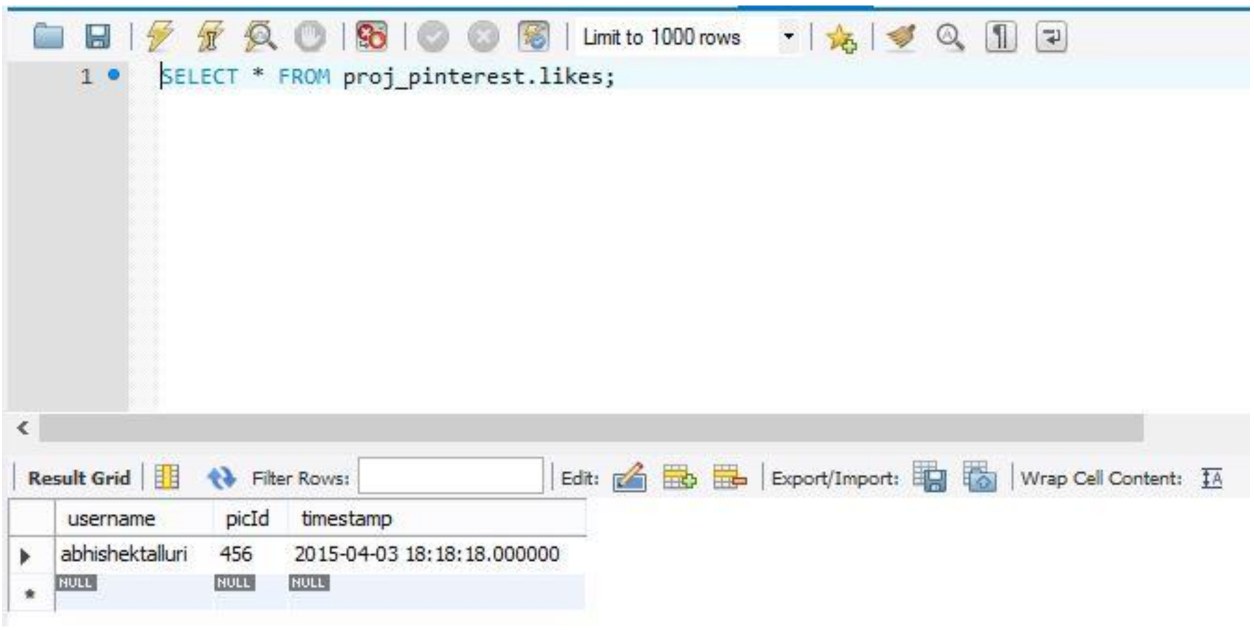
6. Follow Stream:



```
SELECT * FROM proj_pinterest.follow_stream;
```

username	username2	boardname2	timestamp
chandugorthi	abhishektalluri	wildlife	2015-04-02 20:20:20.000000
NULL	NULL	NULL	NULL

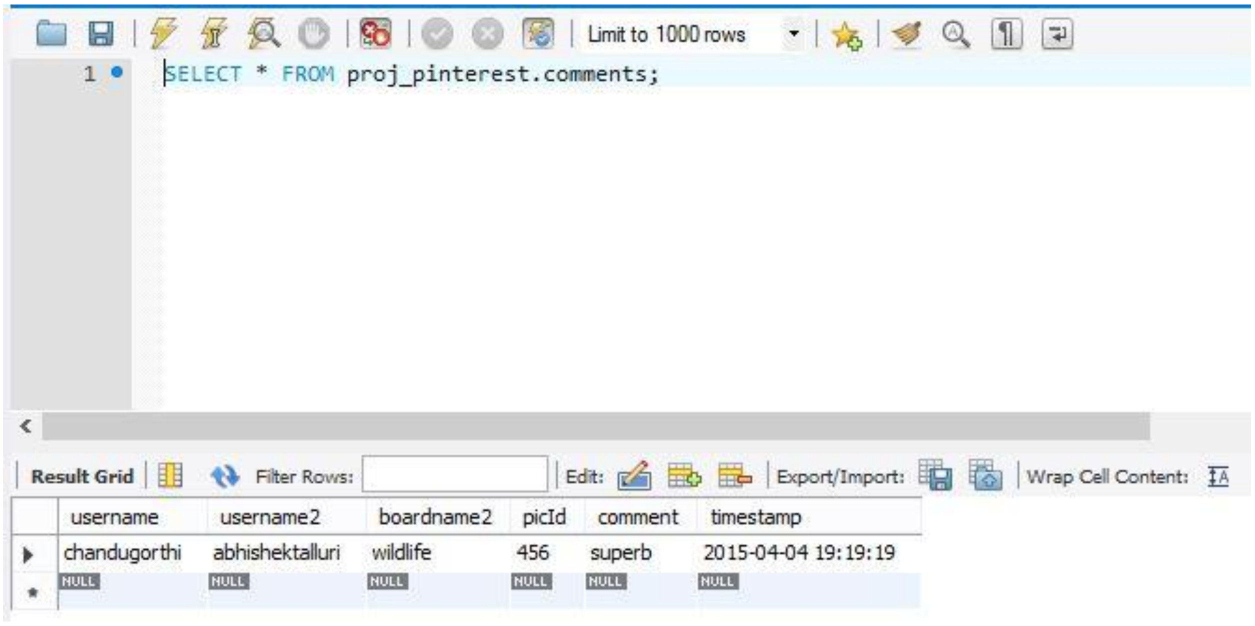
7. Likes:



```
SELECT * FROM proj_pinterest.likes;
```

username	picId	timestamp
abhishektalluri	456	2015-04-03 18:18:18.000000
NULL	NULL	NULL

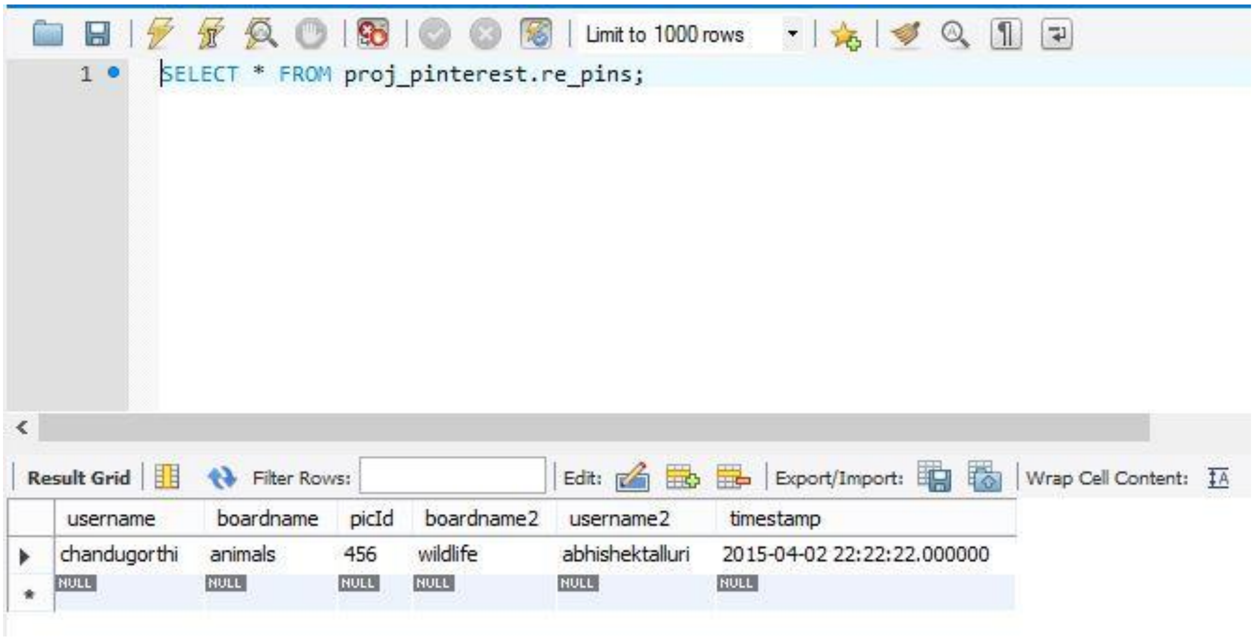
8. Comments:



```
SELECT * FROM proj_pinterest.comments;
```

username	username2	boardname2	picId	comment	timestamp
chandugorthi	abhishektalluri	wildlife	456	superb	2015-04-04 19:19:19
NULL	NULL	NULL	NULL	NULL	NULL

9. Re_pins:



```
SELECT * FROM proj_pinterest.re_pins;
```

username	boardname	picId	boardname2	username2	timestamp
chandugorthi	animals	456	wildlife	abhishektalluri	2015-04-02 22:22:22.000000
NULL	NULL	NULL	NULL	NULL	NULL

Functionalities Implemented:

- **Signup for new users.**

```
$conn2 = getMeDB();
$sql2 = "insert into user (username,firstname,lastname,email,password) values(?,?,?,?,?)";
$sql_login = $conn2->prepare($sql2);
$sql_login->bind_param("sssss", $username,$firstname,$lastname,$email,$password);
$sql_login->execute();
```

- **Login for existing users.**

```
$sql = "select username from user where username=? and password=?";
$sql_login = $conn->prepare($sql);
$sql_login->bind_param("ss", $username,$password);
$sql_login->execute();
$sql_login ->store_result();
$count = $sql_login->num_rows;
if($count == 0)
{
    //echo 'no users';
    //$conn->close();
    header('Refresh: 0;url=login.php?error=1');
} else {
    $_SESSION["username"] = $username;
    $conn20 = getMeDB();
    header('Refresh: 2;url=homepage.php');
}
```

- **Edit profile for users.**

```
$conn2 = getMeDB();
$sql2 = "update user set firstname = ?,lastname = ?,password = ? where username = ?";
$sql_login = $conn2->prepare($sql2);
$sql_login->bind_param("ssss", $newfirstname,$newlastname,$newpassword,$username);
$sql_login->execute();
```

- **A user can create pinboards with sharetype to be public or friends.**

```
$sql = "select username,boardname from pinboards where username=? and boardname=?";
$sql_login = $conn->prepare($sql);
$sql_login->bind_param("ss", $username,$boardname);
$sql_login->execute();
$sql_login ->store_result();
$count = $sql_login->num_rows;
if($count == 0)
{     $conn2 = getMeDB();
      $sql2 = "insert into pinboards (username,boardname,sharetype) values(?,?,?)";
      $sql_login = $conn2->prepare($sql2);
      $sql_login->bind_param("sss", $username,$boardname,$sharetype);
      $sql_login->execute();
      header("Refresh: 0;url=homepage.php");
} else {
      header('Refresh: 2;url=homepage.php?error=1');
}
```

- **Upload picture from local system or from URL.**

Uploading a picture for local directory:

```
$target_dir = "uploads/";
if(isset($_FILES['filename']) && $_FILES['filename']['size'] > 0){
      $size=$_FILES['filename']['size'];
      // getting the image info..
      $imgdetails = getimagesize($_FILES['filename']['tmp_name']);
      $mime_type = $imgdetails['mime'];
      // checking for valid image type
      if(($mime_type=='image/jpeg')||($mime_type=='image/gif')){
        // checking for size again
          $filename=$_FILES['filename']['name'];
          $imgData =addslashes (file_get_contents($_FILES['filename']['tmp_name']));
                $name = $_FILES['filename']['name'];
                if (!get_magic_quotes_gpc()) {
                        $name = addslashes($name);
                }
                $name = md5($name);
                $name = $name . uniqid($name);
                $path = $_FILES['filename']['name'];
                $ext = pathinfo($path, PATHINFO_EXTENSION);
                $name = $name.".".$ext;
                $tag = $_POST['tags'];
                $loc = $_SERVER['HTTP_ORIGIN']."/pin2it/uploads/".$name;
                $target_file = $target_dir . $name;
                if (move_uploaded_file($_FILES['filename']['tmp_name'], $target_file)) {
                        //echo "File is valid, and was successfully uploaded.\n";
                } else {
                        echo "Possible file upload attack!\n";
                }
                //$null = '';
                //$conn1 = getMeDB();
                $sql="INSERT into pictures(picId,username,picture,tags,url) values
                ('','$_SESSION['username']','$imgData','$tag', '$loc')";
                //$sql_pictures = $conn1->prepare($sql);
                //$sql_pictures->bind_param("isss", $null,$imgData,$tag,$loc);
                //$sql_pictures->execute();
                mysql_query($sql,$link) or die(mysql_error());
                header("Refresh: 0;url=uploadphoto.php");
        } else {
          header("Refresh: 0;url=uploadphoto.php?error=1");
        }
} else {
        header("Refresh: 0;url=uploadphoto.php?error=2");
}
```

Uploading a picture from URL:

```
$conn = getMeDB();
session_start();
$username = $_SESSION["username"];
$target_dir = "uploads/";
if(isset($_POST['url'])){
    $url = $_POST['url'];
    $name = basename($url);
    $name = md5($name);
    $name = $name . uniqid($name);
    $ext = pathinfo($url, PATHINFO_EXTENSION);
    $name = $name.".".$ext;
    $loc = $_SERVER['HTTP_ORIGIN']."/pin2it/uploads/".$name;
    $target_file = $target_dir . $name;
    $imgData =addslashes (file_get_contents($url));
    file_put_contents("uploads/$name",file_get_contents($url));
    $tag = $_POST['tags'];
    $sql="INSERT into pictures(picId,picture,tags,url) values ('','$imgData','$tag', '$loc')";
    mysql_query($sql,$link) or die(mysql_error());
    $sql1='SELECT picId from pictures WHERE url = "'.$loc.'"';
    $query1 = mysql_query($sql1) or die(mysql_error());
    $result=mysql_fetch_array($query1);
    $sql = "select picId,username,boardname from pins where username=? and boardname=? and
picId =?";
    $sql_login = $conn->prepare($sql);
    $sql_login->bind_param("sss", $result['picId'],$username,$POST['boardname']);
    $sql_login->execute();
    $sql_login ->store_result();
    $count = $sql_login->num_rows;
    if($count == 0)  {
        $conn2 = getMeDB();
```

```
            $sql2 = "insert into pins (picId,boardname,username,time) values(?,?,?,sysdate())";
            $sql_pins = $conn2->prepare($sql2);
            $sql_pins->bind_param("iss", $result['picId'],$_POST['boardname'],$username);
            $sql_pins->execute();
            header("Refresh: 0;url=homepage.php");
    } else {
            header('Refresh: 2;url=pictures.php?error=5');
    }
}else{
    header("Refresh: 0;url=uploadphotoweb.php?error=1");
}
```

- **Pin pictures from the collection of pictures he has uploaded into his collection or into his other boards.**

```
session_start();
$username = $_SESSION["username"];
$boardname = $_POST['boardname'];
$picId = $_POST['picId'];

$sql = "select picId,username,boardname from pins where username=?
and boardname=? and picId =?";
$sql_login = $conn->prepare($sql);
$sql_login->bind_param("sss", $picId,$username,$boardname);
$sql_login->execute();

$sql_login ->store_result();
$count = $sql_login->num_rows;
if($count == 0)
{
  $conn2 = getMeDB();
  $sql2 = "insert into pins
(picId,boardname,username,time)values(?,?,?,sysdate())";
  $sql_pins = $conn2->prepare($sql2);
  $sql_pins->bind_param("iss", $picId,$boardname,$username);
  $sql_pins->execute();
  header("Refresh: 0;url=homepage.php");
}
else
{
  header('Refresh: 2;url=pictures.php?error=5');
}
```

- **He can re-pin pictures from boards of other users.**

```
session_start();
$username = $_SESSION["username"];
$boardname = $_POST['boardname'];
$boardname2 = $_POST['boardname2'];
$otheruser = $_POST['otheruser'];
$picId = $_POST['picId'];
$sql = "select picId,username,boardname from pins where username=? and boardname=? and picId
=?";
$sql_login = $conn->prepare($sql);
$sql_login->bind_param("sss", $picId,$username,$boardname);
$sql_login->execute();
$sql_login ->store_result();
$count = $sql_login->num_rows;
if($count == 0) {
    $conn2 = getMeDB();
    $sql2 = "insert into pins (picId,boardname,username,time) values(?,?,?,sysdate())";
    $sql_pins = $conn2->prepare($sql2);
    $sql_pins->bind_param("iss", $picId,$boardname,$username);
    $sql_pins->execute();
    $conn3 = getMeDB();
    $sql3 = "insert into repins (username,boardname,picId,boardname2,username2,time)
values(?,?,?,?,?,sysdate())";
    $sql_repins = $conn3->prepare($sql3);
    $sql_repins->bind_param("ssiss", $username,$boardname,$picId,$boardname2,$otheruser);
    $sql_repins->execute();
    header("Refresh : 0;url =board.php?otheruser=$otheruser&boardname=$boardname2");
}
```

- **Create follow streams.**

```
session_start();
$username = $_SESSION["username"];
$streamname = $_POST['streamname'];
```

```
$sql = "select username,streamname from streams where username=? and streamname=?";
$sql_stream = $conn->prepare($sql);
$sql_stream->bind_param("ss", $username,$streamname);
$sql_stream->execute();
$sql_stream ->store_result();
$count = $sql_stream->num_rows;
if($count == 0) {
    $conn2 = getMeDB();
    $sql2 = "insert into streams (username,streamname) values(?,?)";
    $sql_login = $conn2->prepare($sql2);
    $sql_login->bind_param("ss", $username,$streamname);
    $sql_login->execute();
    header("Refresh: 0;url=homepage.php");
}
```

- **A user can follow boards of other users by adding that board to one of his follow streams.**

```
session_start();
$username = $_SESSION["username"];
$boardname = $_POST['hiddenBoardId1'];
$otheruser = $_POST['hiddenUser'];
$streamname = $_POST['streamname'];
$conn = getMeDB();
$sql = "insert into followstreams (username,streamname,username2,boardname2,time)
values(?,?,?,?,sysdate())";
$sql_followstreams = $conn->prepare($sql);
$sql_followstreams->bind_param("ssss", $username,$streamname,$otheruser,$boardname);
$sql_followstreams->execute();
header("Refresh: 0;url=homepage.php");
```

- **A user can search for other users and can see his boards.**

```
$conn1 = getMeDB();
$srchString = $_POST['key'];
$wild="%$srchString%";
$sql2 = 'select * from user where username LIKE "' . $wild . '"';
$result = $conn1->query($sql2);
$userslist = [];
while($row = $result->fetch_row())
{
    $userslist[] = $row;
}
$usersencoded = json_encode($userslist);
print_r($usersencoded);
```

- **A user can also search for pics with a particular tag that he wants.**

```
session_start();
$username = $_SESSION['username'];
$conn1 = getMeDB();
$srchString = $_POST['key'];
$wild="%$srchString%";
$sql2 = 'select url from pictures where username = "'.$username .'" and tags LIKE "' . $wild . '"' ;
$result = $conn1->query($sql2);
$picIdList = [];
//$totalStringsThis = '';
while($row = $result->fetch_row())
{
    $picIdList[] =$row[0];
}
$jsonEncoded = json_encode($picIdList);
print_r($jsonEncoded);
```

- **A user can send friend requests to other users.**

```
session_start();
$username = $_SESSION["username"];
$friend = $_POST['otherUser'];
$conn1 = getMeDB();
$status = 'request sent';
$sql_friends = "insert into friends (username,username2,status,time) values (?,?,?,sysdate())";
$sql_prep_friends = $conn1->prepare($sql_friends);
$sql_prep_friends->bind_param("sss", $username,$friend,$status);
$sql_prep_friends->execute();
```

- **A user can accept or decline friend request from other users.**

```
$conn1 = getMeDB();
$sql_friends = "UPDATE friends SET status = ?, time = ? WHERE username = ? AND username2 = ?";
$sql_prep_friends = $conn1->prepare($sql_friends);
$time = date("Y-m-d h:i:sa");
if(isset($_POST['accept'])){
    $status = 'accepted';
} else {
    $status = 'rejected';
}
$sql_prep_friends->bind_param("ssss", $status,$time,$friend,$username);
$sql_prep_friends->execute();
header("Refresh: 0;url=homepage.php");
```

- **When a user opens one of his follow streams, he can see all the pictures from the boards he added to that follow stream.**

- **Homepage contains the pictures from his follow streams and collection of pictures he has in his account.**
- **A user can like his pics and other users pics**

```
session_start();
$username = $_SESSION["username"];
$sql = "insert into likes (username,picId,time) values(?,?,sysdate())";
```

```
$sql_likes = $conn->prepare($sql);

$sql_likes->bind_param("si", $username,$picId);

$sql_likes->execute();
```

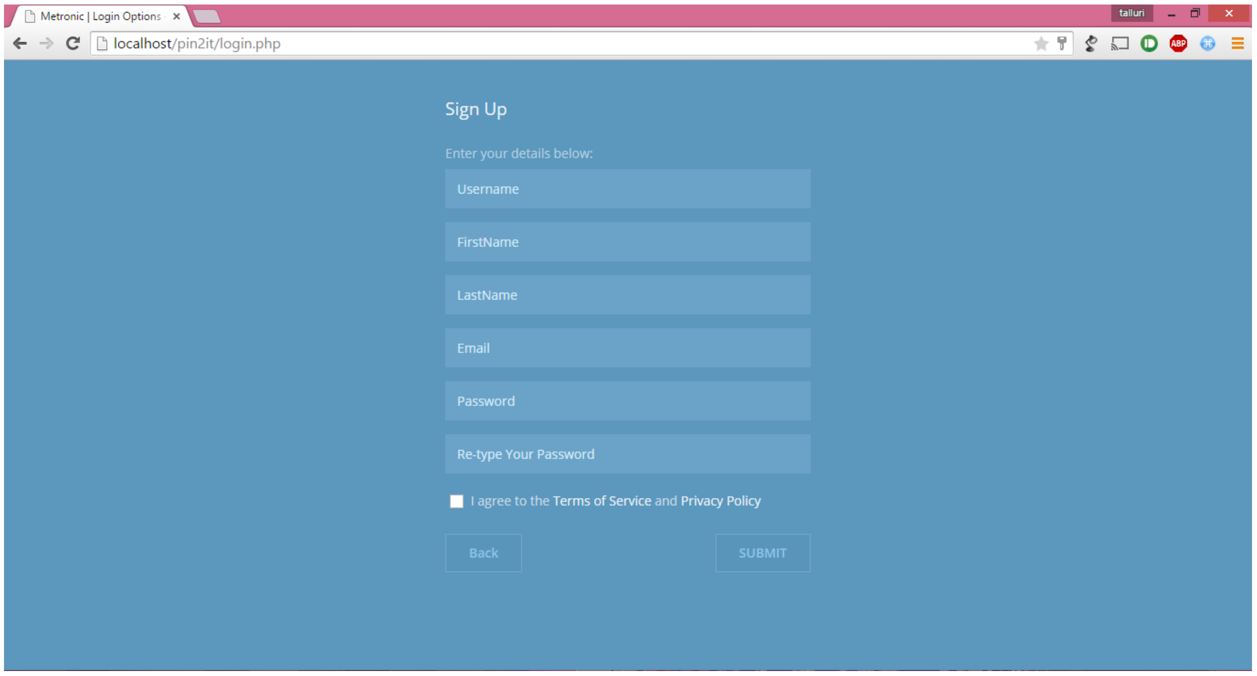
- **He can comment on his pics.**

```
$sql = "insert into comments (username,username2,boardname2,picId,comment,time)
values(?,?,?,?,?,sysdate())";

$sql_comments = $conn->prepare($sql);

$sql_comments->bind_param("sssis", $username,$otherUser,$boardname,$picId,$comment);

$sql_comments->execute();
```

- **If a user has marked his board's sharetype to be friends, then only his friends can comment on the pics in that board**

Screenshots:

- Signup for new users.


Metronic | Login Options

localhost/pin2it/login.php

Sign Up

Enter your details below:

Username

FirstName

LastName

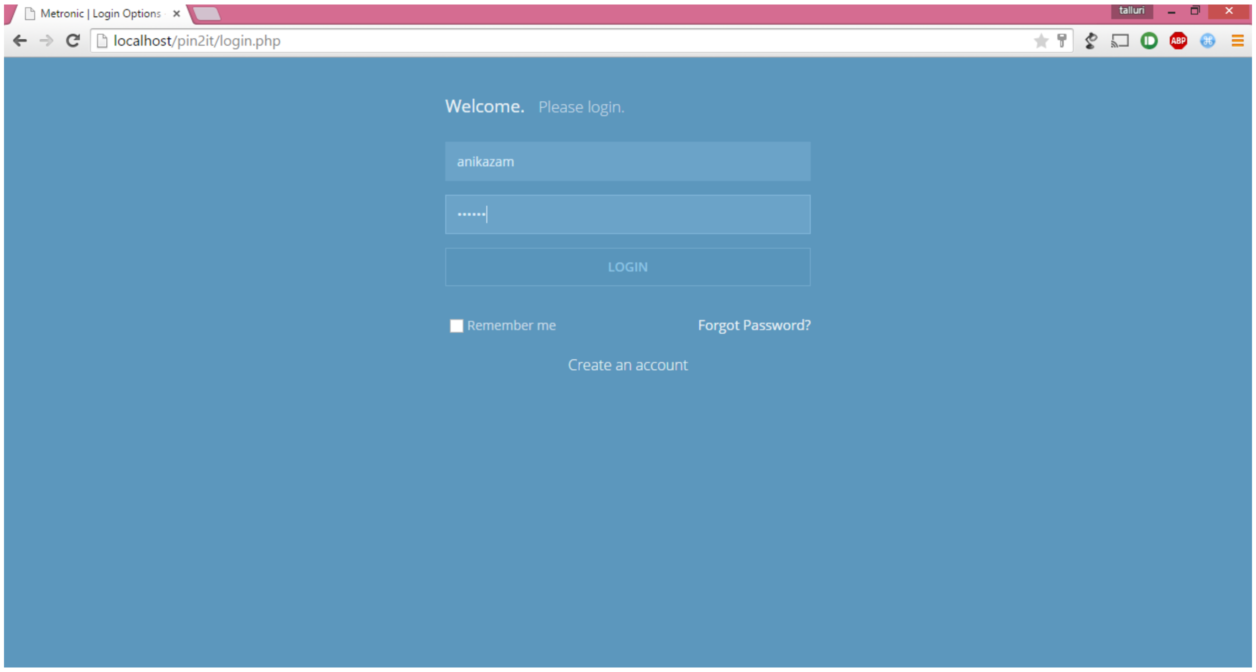
Email

Password

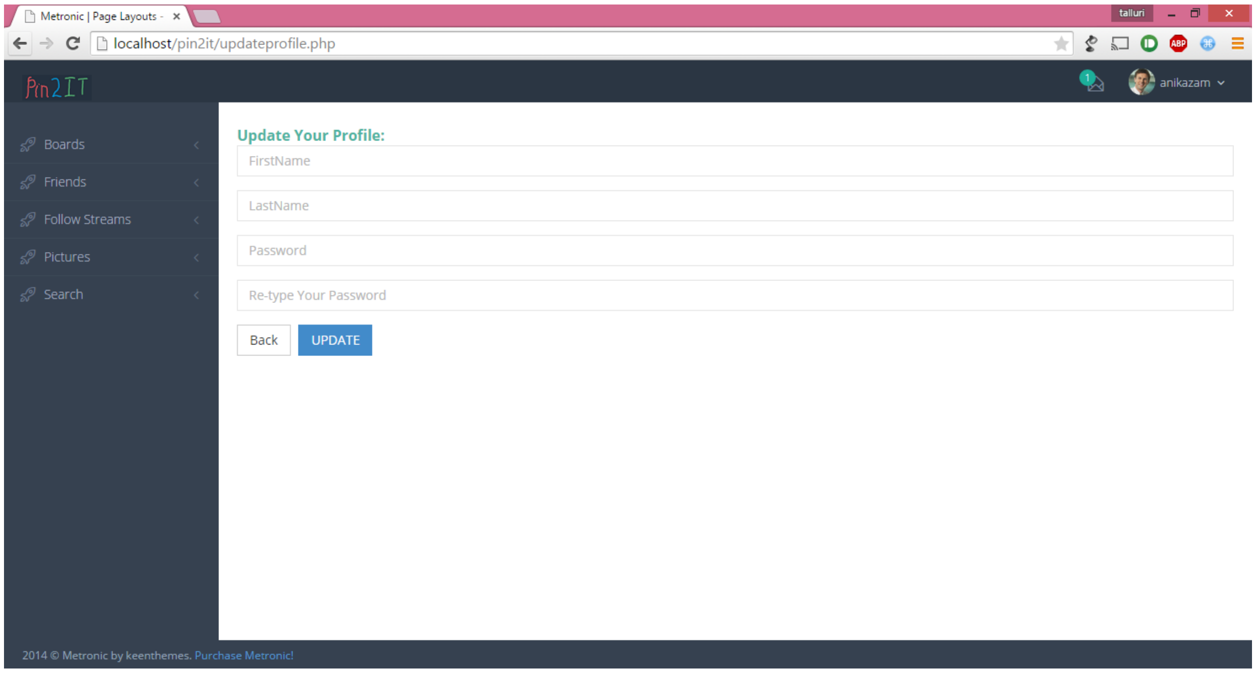
Re-type Your Password

I agree to the Terms of Service and Privacy Policy

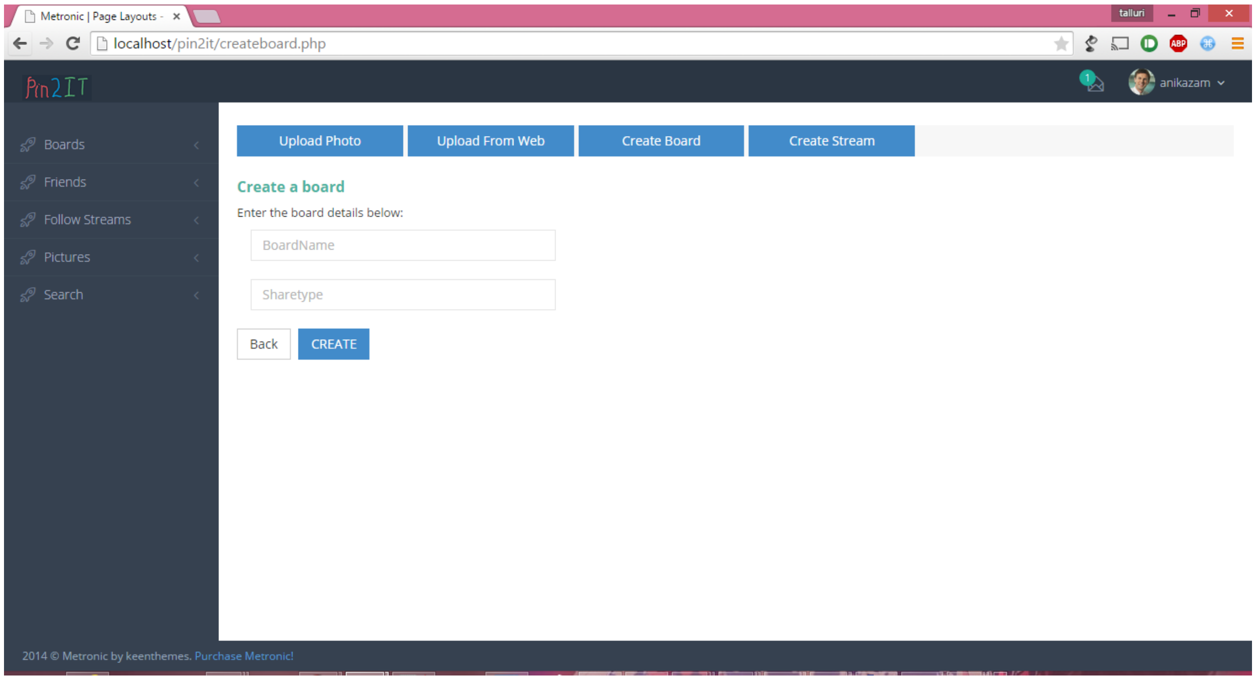
- Login for existing users.



- Edit profile for users.



- A user can create pinboards with sharetype to be public or friends.



Create a board

Enter the board details below:

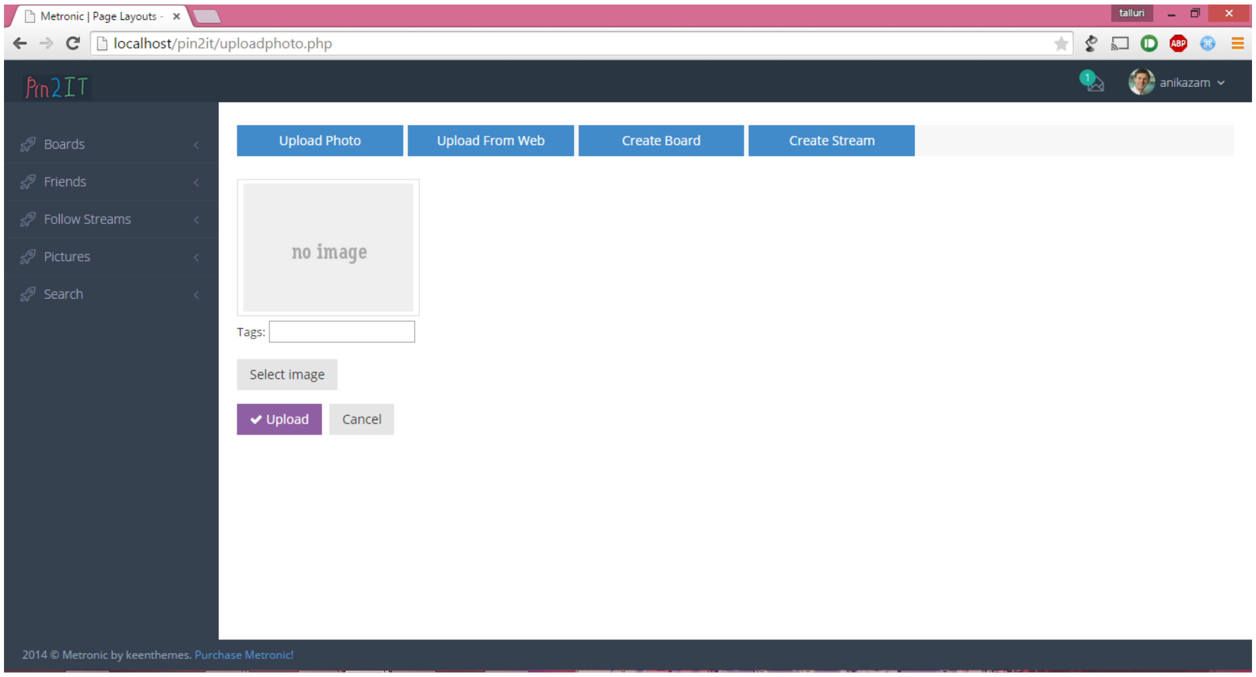
BoardName

Sharetype

Back CREATE

2014 © Metronic by keenthemes. Purchase Metronic!

- Upload picture from local system or from URL.

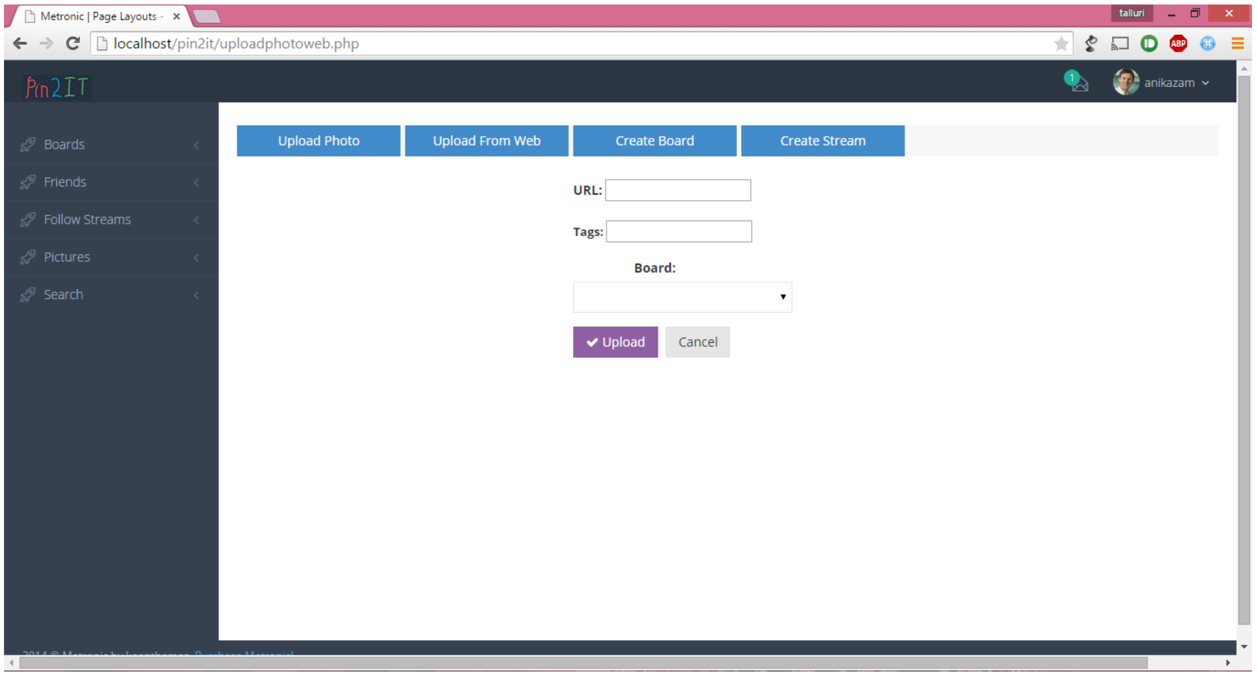


Tags:

Select image

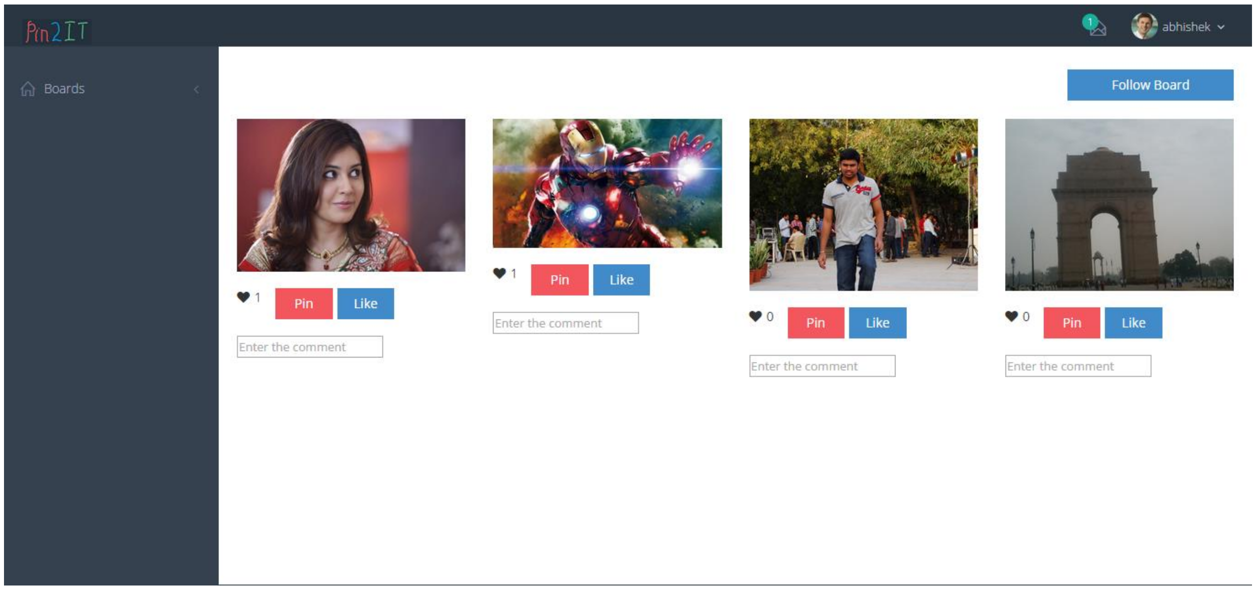
Upload Cancel

2014 © Metronic by keenthemes. Purchase Metronic!

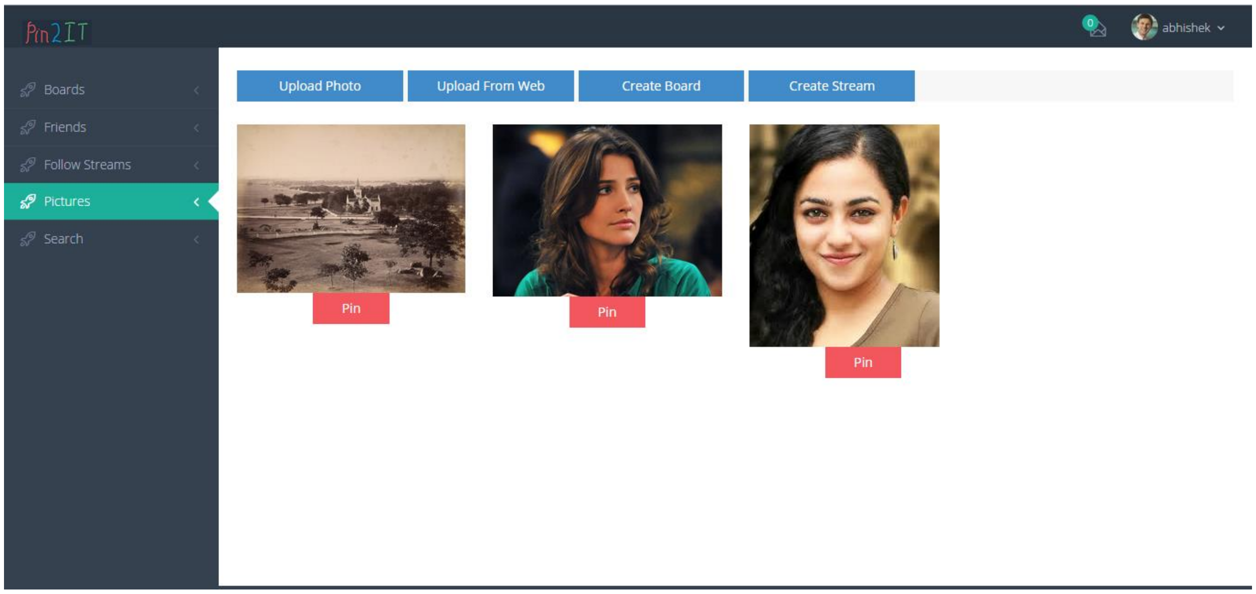


- Pin pictures from the collection of pictures he has, his other boards or from boards of other users.

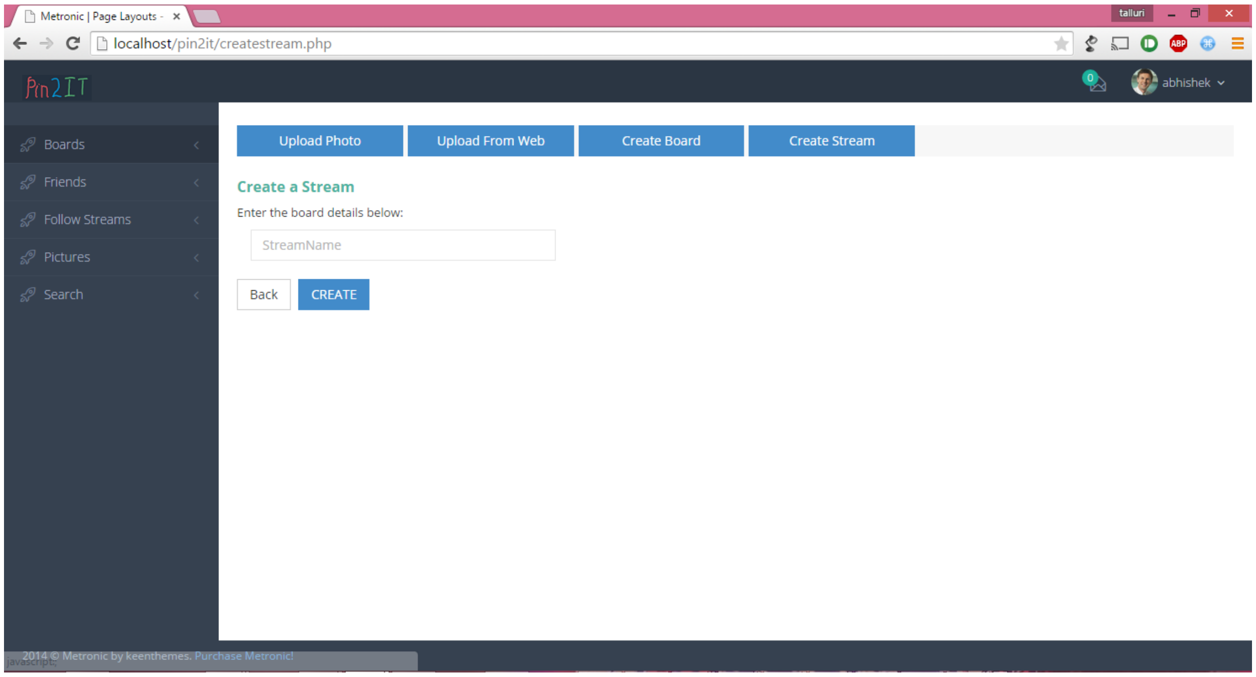
Other User Board:



From collection of pictures of same user:



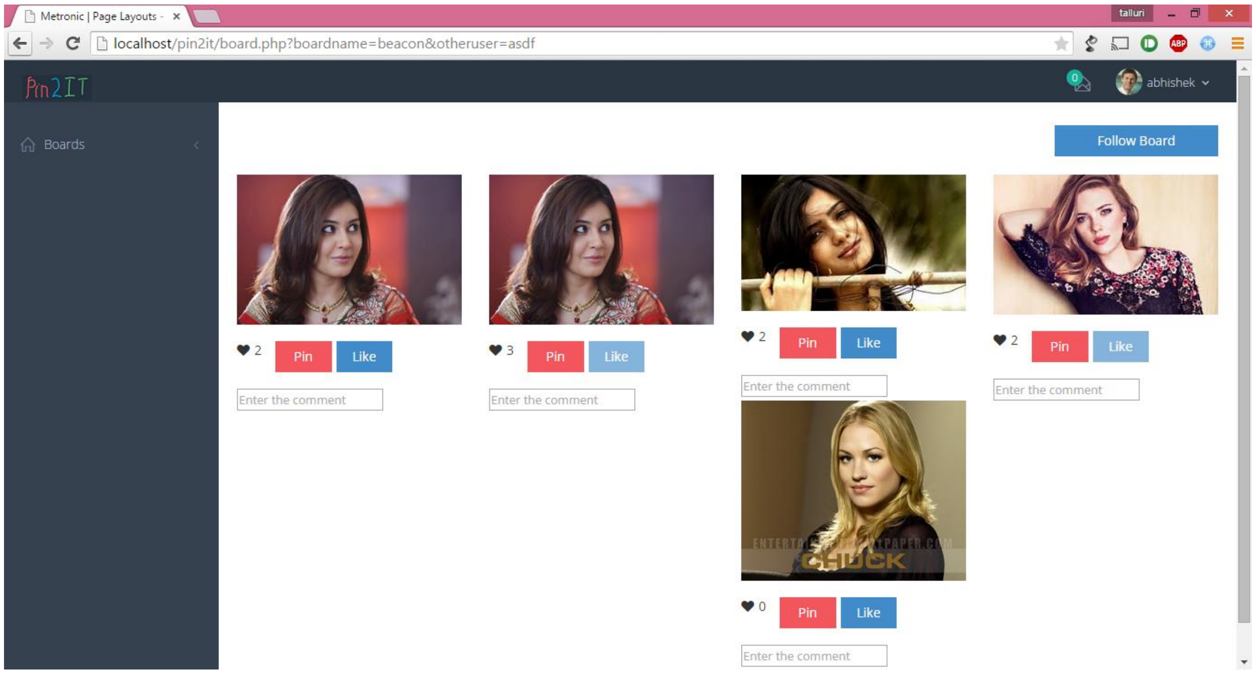
- Create follow streams.



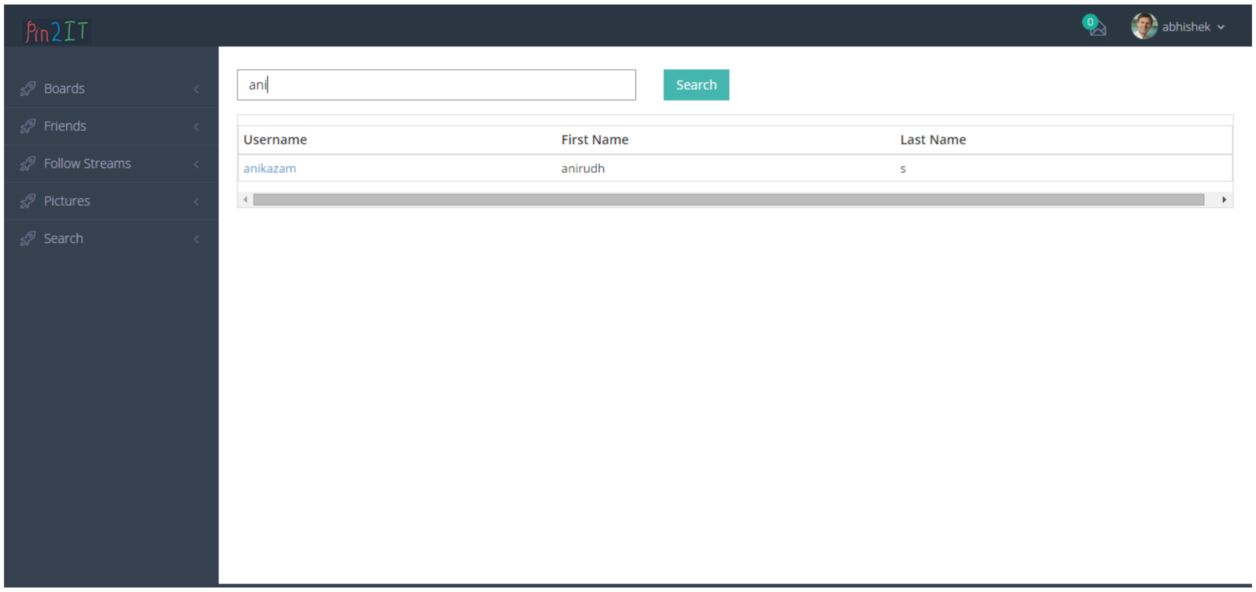
Create a Stream

Enter the board details below:

- **A user can follow boards of other users by adding that board to one of his follow streams.**

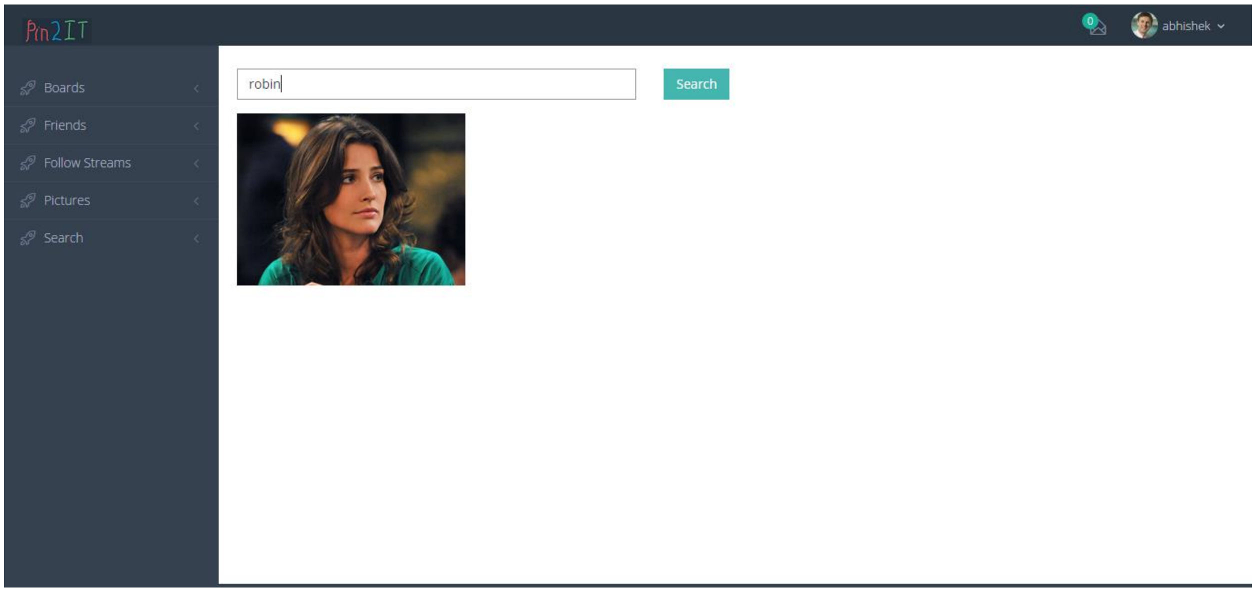


- **A user can search for other users and can see his boards.**

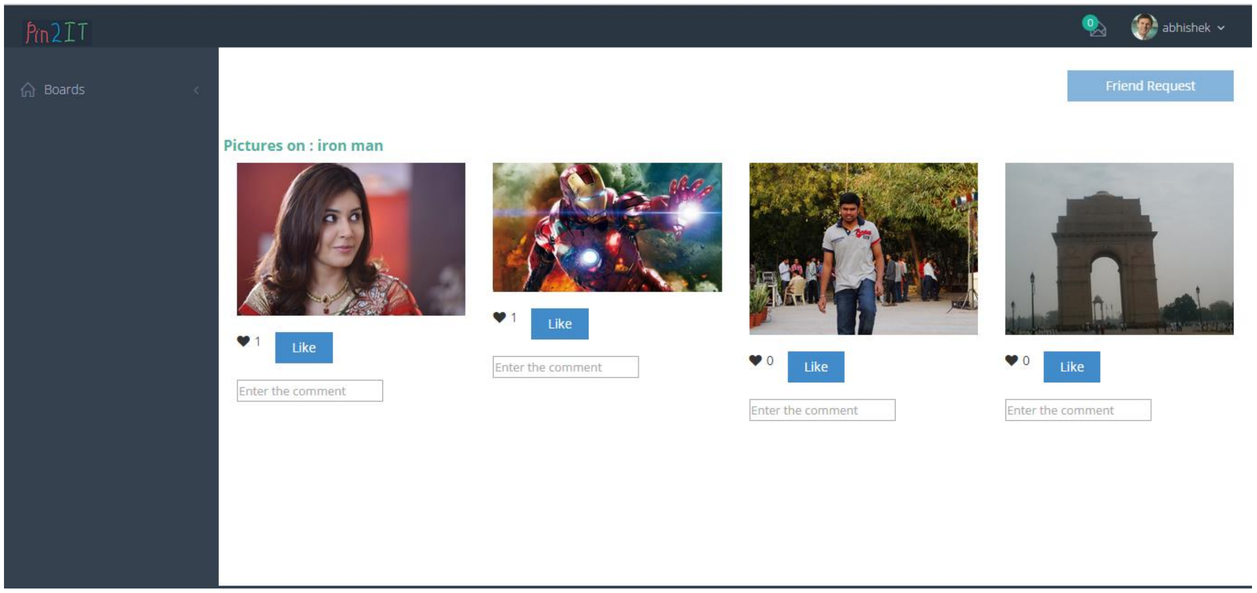


Username	First Name	Last Name
anikazam	anirudh	s

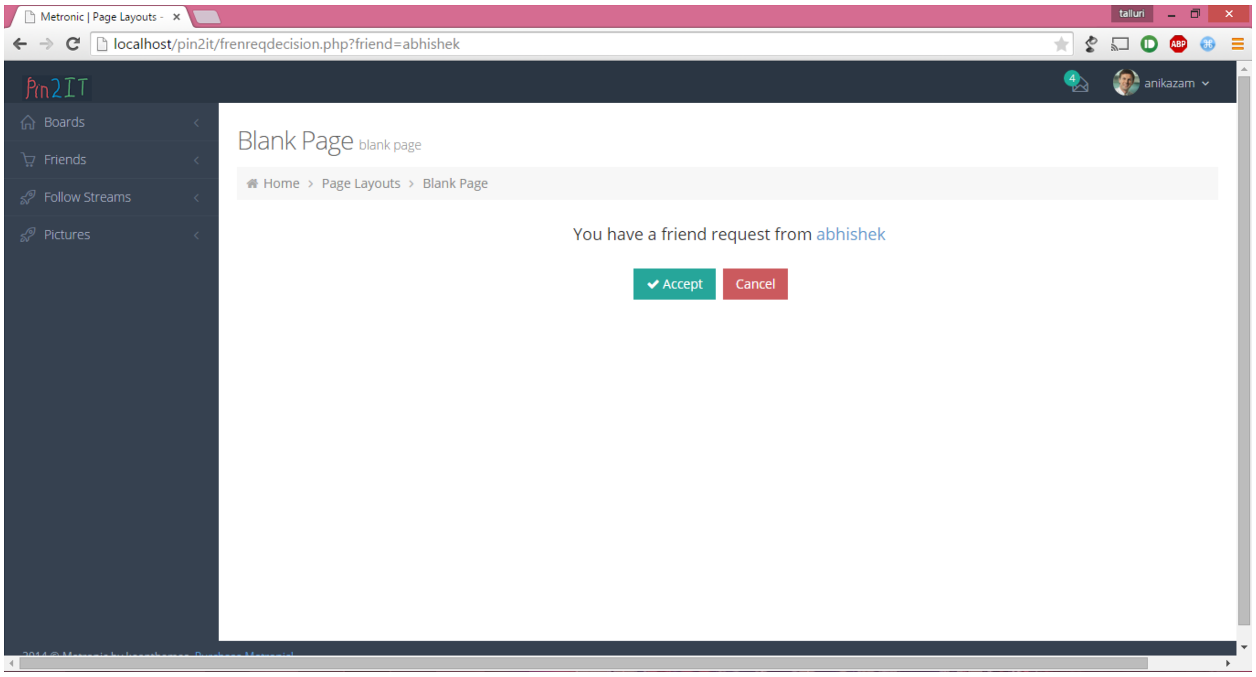
- **A user can also search for pics with a particular tag that he wants.**



- **A user can send friend requests to other users.**

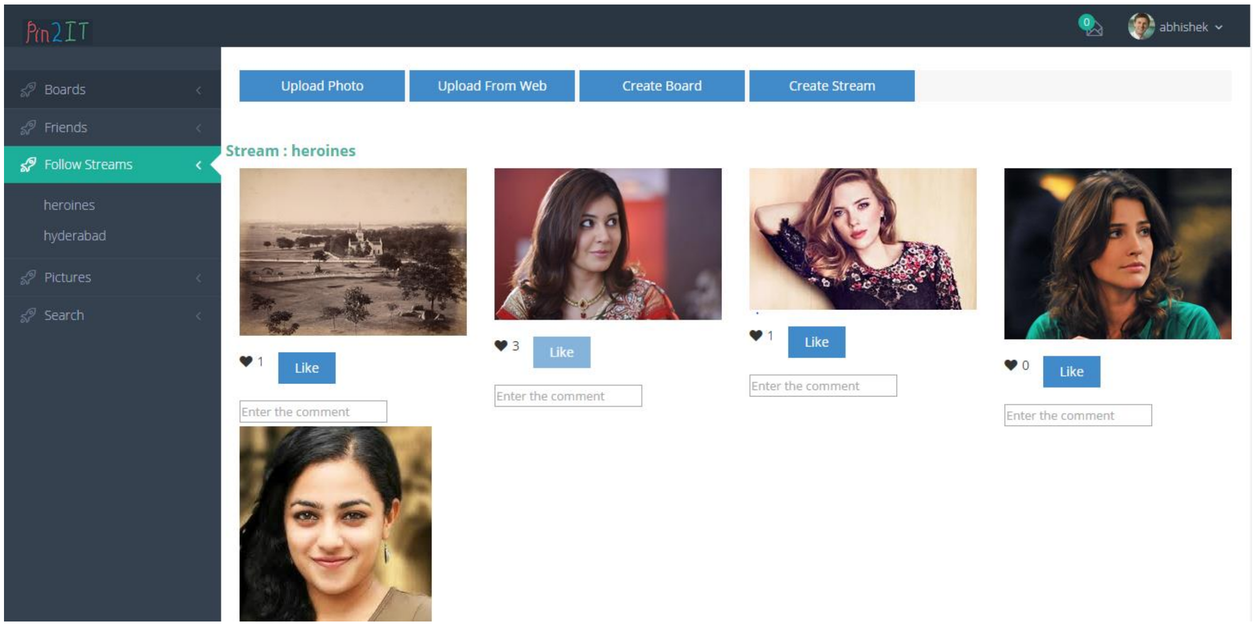


- **A user can accept or decline friend request from other users.**

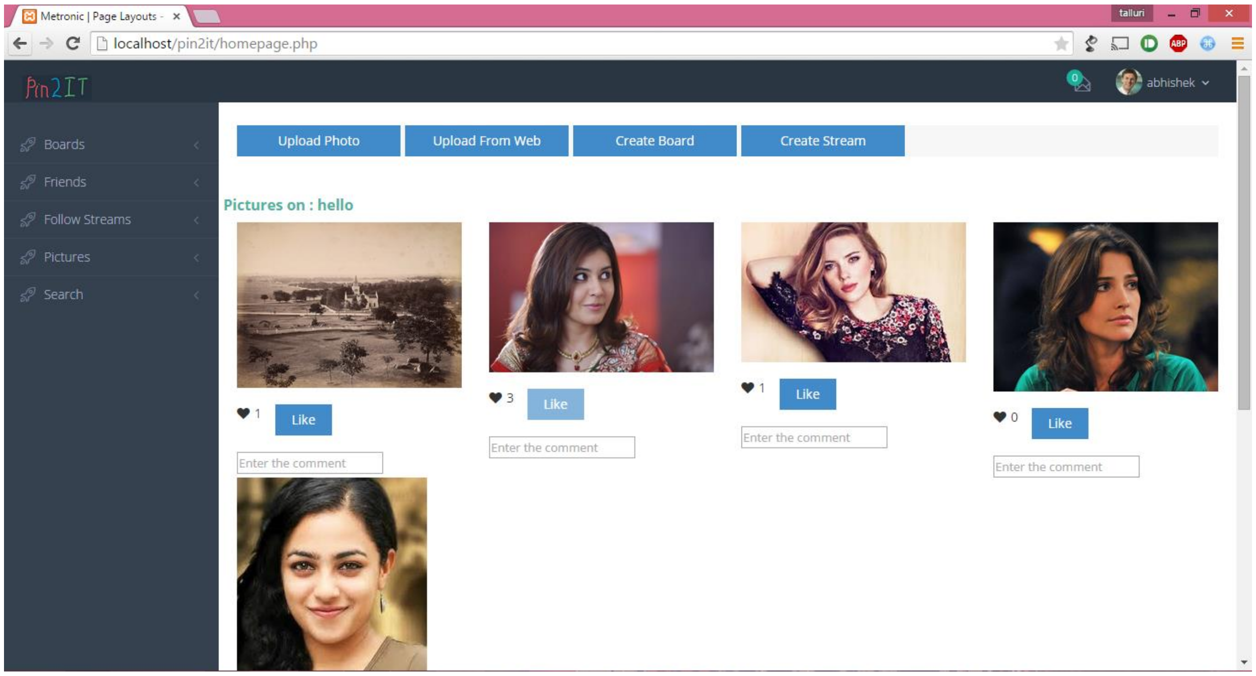

Metronic | Page Layouts -
localhost/pin2it/frenreqdecision.php?friend=abhishek
Pin2IT
anikazam
Boards
Friends
Follow Streams
Pictures
Blank Page blank page
Home > Page Layouts > Blank Page

You have a friend request from abhishek

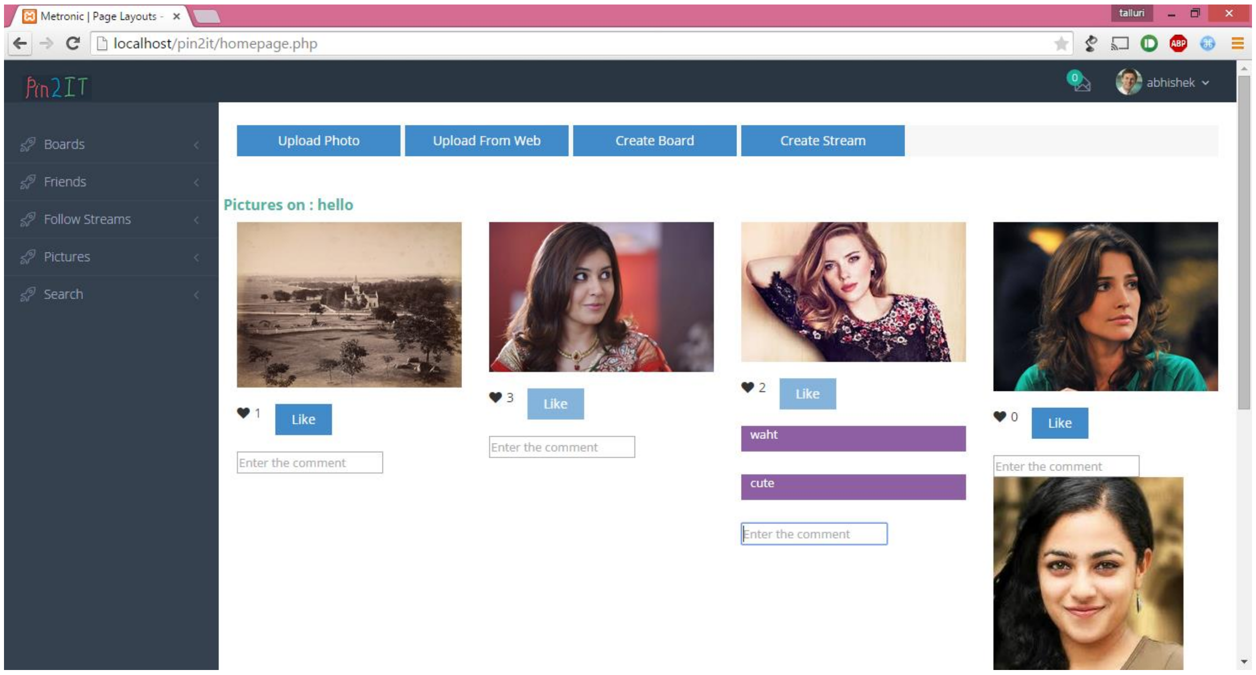
- **When a user opens one of his follow streams, he can all the pictures from the boards he added to that follow stream.**



- **Homepage contains the pictures from his follow streams and collection of pictures he has in his account.**



- **A user can like and comment on his pics.**


Metronic | Page Layouts -
talluri
localhost/pin2it/homepage.php
Pin2IT
abhishek
Boards
Friends
Follow Streams
Pictures
Search
Upload Photo
Upload From Web
Create Board
Create Stream
Pictures on : hello
1
Like
Enter the comment
3
Like
Enter the comment
2
Like
waht
cute
Enter the comment
0
Like
Enter the comment