tripNote Assignment4 report.

Marcus Su
301113871

introduction

Planning a trip may be painful for many people, buy fly tickets, book the hotels, figure out local traffic, and plan the route, etc. all the things have to be done ahead of schedule. The tripNote is design for solve this problem. Experienced traveller can post their trip journal on the website, the journal may include hotel, restaurant they have been, some goodplace worth to go, and other experiences. For the normal user, they can search the trip journal by setting up the key word like departure and destination. The idea is to let busy people spend less time on planning the trip, and let experienced traveller to share their experience to others.

Functions, features

The user can register as a member or a visitor, the difference is that a bollean value can decide whether or not they are allow to write note.

The visitor/member can follow/unfollow many other users, since visitors are not allowed to write note, there is not way to follow them.

The user (member only) can write new note, delete, or edit a existed note.

All user can use the search tool in home page. The searching is depends on three values, departure, arrive, season, user can leave the dropdown menu as defult for all search values, for example, a selection like this will only search any tripNote departure from London.

Database

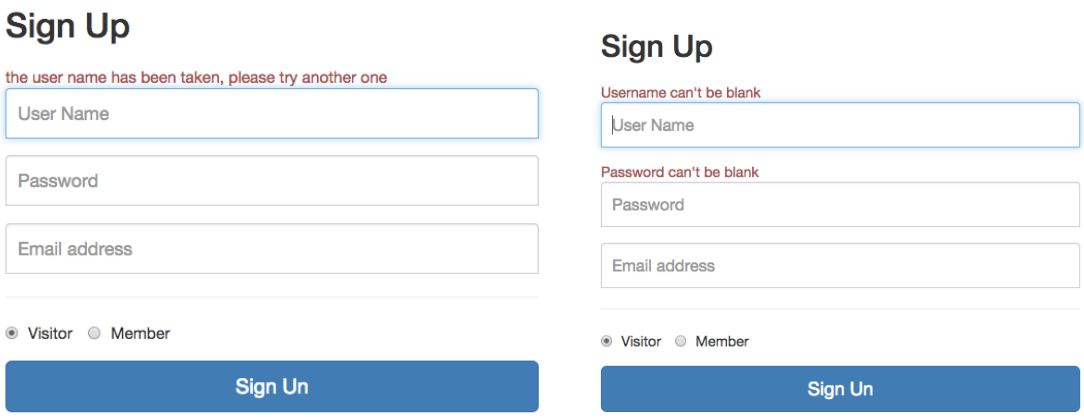

```
DROP TABLE IF EXISTS `note`;
CREATE TABLE IF NOT EXISTS `note` (
`id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `textarea` text,
  `departure` varchar(50) NOT NULL,
  `arrive` varchar(50) NOT NULL,
  `season` varchar(50) NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=lat

DROP TABLE IF EXISTS `twitter`;
CREATE TABLE IF NOT EXISTS `twitter` (
`id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL,
  `profile_image_url` text,
  `textarea` text,
  `postedtime` varchar(50) NOT NULL,
  `time` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=lat

DROP TABLE IF EXISTS `flickr`;
CREATE TABLE IF NOT EXISTS `flickr` (
`id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `farm` text,
  `server` text,
  `photoid` text,
  `secret` varchar(50) NOT NULL,
  `time` varchar(50) NOT NULL
```

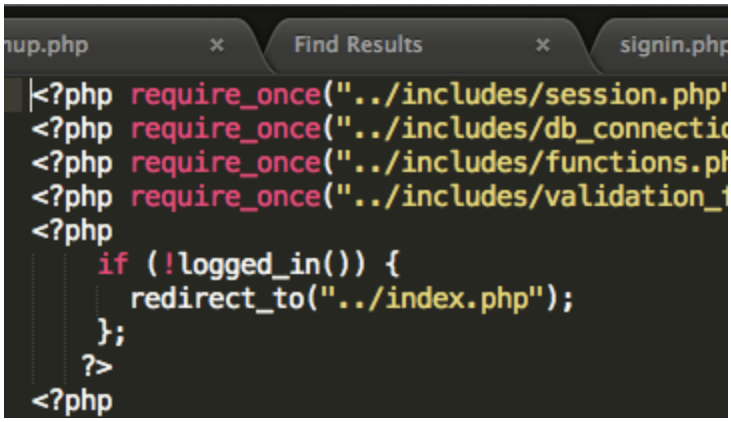
The database contains 5 tables, each table has foreign key to connect with other tables, for example use the user name ‘Marcus’ as a index to search on both flickr and note tables to get photos and notes.

Security



Secure authentication is applyed to prevent user input blank, malicious code or register the same username.

The webside saves user's login information in the $_SESSION and clear it up when user log out. If a user try to access some page without providing log in information, page will be redirected to index.php

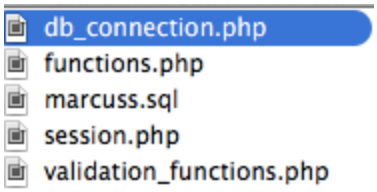


user's password are encrypted using blowfish library.

hashed_password
$2y$10$ZjhhN2VkZjIxZGVhMWZiOOEeLuZW0rG.IhFB1Y6CXLh...
$2y$10$ZTAzMzUwZTI1ZTZiNzdlOORw6tPd5iyJOAQjHJiLkM9...
$2y$10$MzZjMjE0ZWQ0YTczOGI2Nupbf2FS3pcquvEwhpAfAeJ...
$2y$10$NWVjOTVkNjgzODE1OGE1YuMarRNvLPDlWFR.XUgiV96...
$2y$10$MjcxZDljYjhiMDZjYTEwOOUY7CyhJrn3kyyt81Uwzat...
$2y$10$YmQ1OWI3NzgzMTIxZjYyMeiy8Wt3HEPUO14MVw4pc.H...
$2y$10$NDYwZDg4MmZlMzU4MWU3ZOCS0wbkMQS/QQG3VGdDAfe...
$2y$10$NzUyOWJlMWE3ODBkMTU1O.Ecugb3ayD8B.3jsbOKQkI...
$2y$10$MTc0MjIxZDgxZmZmZmVjZ.yu7oXoJ1AjisqaXR328ya...
$2y$10$MjdjNTMwZWFiZjE4MGI5MuD3LUmgzCMn4OpQ76H48Ne...
$2y$10$MGQzNzliMTY2M2M3ODUxYuic4b/NqoBFr1JxYlhLh9J...

Clarity and modularization

For convenience, and maintenance, reuseable functions are put into couple files, and load when need,



db_connection.php contains all the information to access to the data base.

functions.php contains some main algorithms.

marcuss.sql is the exported database file.

session.php is loaded on every page to store log in information.

validation_functions.php contains functions validate user's input.

Twitter and Flickr

The webside allows user to enter their social account for other user to follow, I added two tables "twitter" and "flickr" to store user's social media information in the database. User will be asked to input their twitter and flickr account name when doing the registration, if the input is blank or invalided, it will not be stored, if yes, store all the necessary information with a time stamp, when the next time user request to see it, check the current time and time stamp, if it's more than 60 sec, update the twitter timeline/flickr image link, and display it, if it's less than 60 sec, get the information form database.

AJAX

The AJAX is applied to keep the website update without refreshing the whole page. It contains two parts,

```
//output the lastest tweets any way.

    $name=$tweets['0']['user']['name'];
    $profile_image_url=$tweets['0']['user']['profile
    $i = 0;
    header ("Content-Type:text/xml");
     echo "<?xml version='1.0' encoding='UTF-8'?>";
     echo "<tweets>";
    for($x = 0; $x < $length; $x++){
    $twitter_text=addslashes($tweets[$x]['text']);
    $postedtime=$tweets[$x]['created_at'];
    echo "<item>
            <username>".stripslashes($twitter_name)."<
            <name>".stripslashes($name)."</name>
            <profile_image_url>".$profile_image_url."<
            <twitter_text>".stripslashes($twitter_text
            <postedtime>".$postedtime[0].$postedtime[1
            </item>";
    }
    echo "</tweets>";
}
```

This XML file does not appear to have any s

```
▼<tweets>
  ▼<item>
      <username>marcus</username>
      <name>Marcus □</name>
    ▼<profile_image_url>
        http://pbs.twimg.com/profile_images/5663163
      </profile_image_url>
    ▼<twitter_text>
        4h 18m 20s. Done. Next challenge? I just ran
      </twitter_text>
      <postedtime>Sun Mar 15</postedtime>
    </item>
  ▼<item>
      <username>marcus</username>
      <name>Marcus □</name>
    ▼<profile_image_url>
        http://pbs.twimg.com/profile_images/5663163
```

PART1: the "get_tweets.php" file requires user's tweets form twitter's API and out put a XML files.

```
function tweetsData() {

    $.ajax({

     url: "get_tweets.php",
     dataType: "xml",
     success: function(data) {

      $("#twitter").children().remove();

      $(data).find("item").each( function() {

       var info = "<div class=\"box row marginleft marginright\"><div
       $("#twitter").append(info);

      });

     },
     error: function() { $("#twitter").children().remove();
                         $("#twitter").append("<p>There was an er
    });
}
```

PART2: in “refresh_tweets.js” file, call “get_tweets.php” to get the xml file, and use DOM API to select contents form xml, format them and insert into html files, this function will call every **10sec** to check if there is new contents and update it(notes and flickr use the same method to get update)