|  |  |
| --- | --- |
| find your Group  Final Project Report | Abstract The project focused on a full functional social media web application with fancy and precise UIs, which is also published through Amazon EC2. The link is www.findyougruops.com, please feel free to visit.  Fangning He, Fei Li, Yunlu Zhou, Zhiyi Wang  INFO6150 38687 Web Design/User Experience Engr SEC 04 - Spring 2017 |

Contents

[Topic of the project: 2](#_Toc480533541)

[Introduction 2](#_Toc480533542)

[Problem definition 2](#_Toc480533543)

[Scope 2](#_Toc480533544)

[Data flow 2](#_Toc480533545)

[Configuration 3](#_Toc480533546)

[Technologies: 3](#_Toc480533547)

[Project Detail Introduction 3](#_Toc480533548)

[Register 3](#_Toc480533549)

[Login 3](#_Toc480533550)

[User Home 3](#_Toc480533551)

[User Personal Page 4](#_Toc480533552)

[Friend Page 5](#_Toc480533553)

[Photo Wall: 5](#_Toc480533554)

[Profile: 7](#_Toc480533555)

[Database and picture stored configuration 8](#_Toc480533556)

## Topic of the project:

Social Media Web Application

## Introduction

## Problem definition

Using frond end technique to build a social media website, providing an approach for people to share ideas, exchange cultures and connect with others.

## Scope

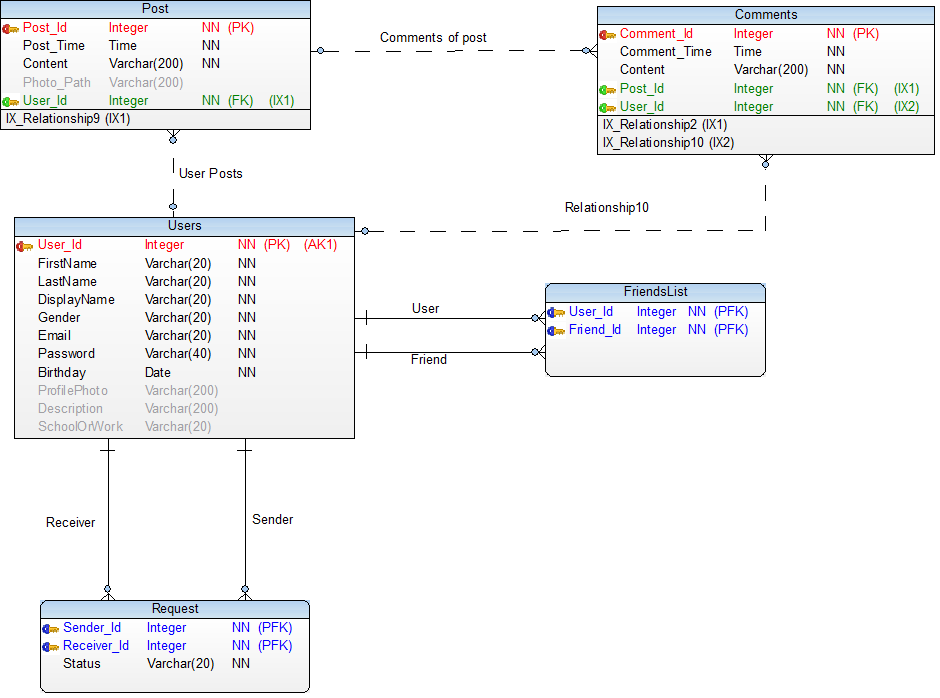
1. Register/Login/Logout
2. Create profile
3. Search a friend
4. Add/Remove a friend
5. Post personal blogs and photos
6. Leave or delete comments

## Data flow

C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram for final.png

As is shown above in the data flow diagram, clients start from Login page, in which if clients have not registered then register first. After registration, users will come into User Home page. The next level is User Personal page, from there, users can visit their Friends` pages. Both in User Home page and Friend page, there are Post, Photo, Profile and Friend List sections. Particularly, we designed Photo Wall page for Photo section.

## Database



Starting with registration, when user registered on our website, his or her information will be stored in table of ‘Users’, then the user can log in. The login page will pull the data from ‘Users’ table to validate this user’s username (email) and password combination.

User can post either text or photos through the user home page, and all the posts will be stored in ‘Post’ table. The friends of this user have ability to comment each post, and the comments will be stored in ‘Comments’ table. The Comments table has foreign keys of both ‘Post\_Id’ and ‘User\_Id’, so we can know which post is the comment under, and which user of this comment belongs to.

The ‘FriendsList’ is nothing but a table to store friends list for a certain user. And ‘Request’ table is for user adding new friend. When a user is trying to add another user, he will the sender to send a friend request, and another user will receive a friend request; as another user approved the request, the ‘Status’ attribute will update and indicate that they are friends now so the program can work on friends list update.

## Configuration

## Technologies:

**Front-end:** HTML5，CSS3, Bootstrap, JavaScript, JQuery

**Back-end**: PHP, SQL

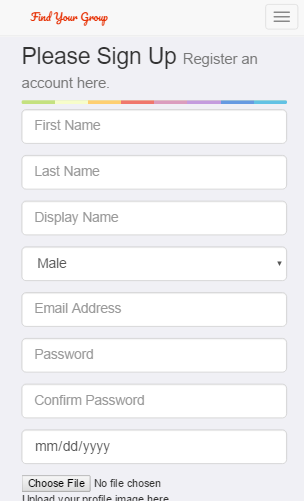
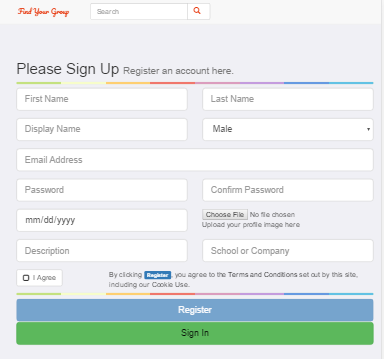
## Project Detail Introduction

1. Register a new user and login to the website.
2. Edit profile
3. Browse all posts from friends, view friends on the sidebar
4. Search users through navigation bar
5. Browse personal page
6. Add new friends
7. Push friends request
8. Delete post on user’s own page
9. Photo wall to display user’s photo album and all the photos he/she has ‘liked’

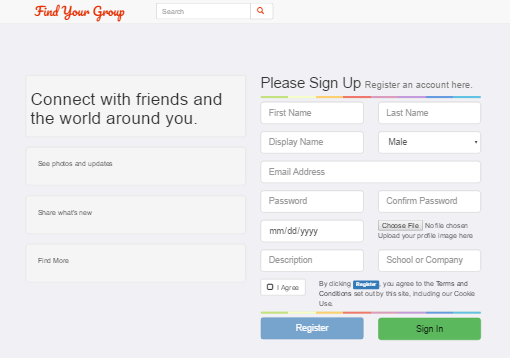
### Register

**Design:**

Register page is where your start to register your account into our social website, which implied a fully responsive, dynamic design for different screen sizes of devices, with both front and back end validations.

Phone Display Tablet Display



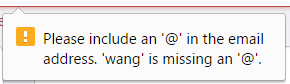
### Login

Laptop/Desktop Display

Inputs explanation: starting with input fields as picture shown above, ‘First Name’, ‘Last Name’, ‘Display Name’

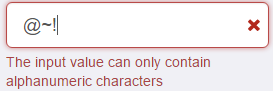
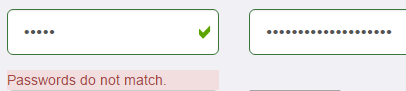
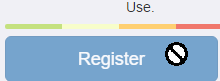
**Three-layers validation:**

The first layer validation is executed by front-end layer, which includes JavaScript, jQuery, and HTML. The HTML will validate inputs based in HTML tags assigned tags, which as we know is not very reliable but makes it easier at certain level.

HTML Validation 1 HTML Validation 2

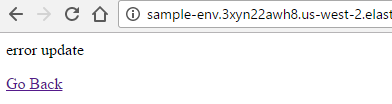
Then input validation will also be done by JavaScript and jQuery. With appreciation for convenience of library and APIs from jQuery, we can validate the blankness, input types, and input formats from the input given by users. For password inputs, we noticed it’s a special case, since the user need to input not only ‘Password’ but also ‘Confirm Password’ field to double check, so the page also validates these, based on JavaScript function we created by our own. So, when two passwords do not match, it will pop up warning for user.

Input Format Validation(JS) Passwords Match Validation (JS) Button Disable Validation(JS)

The second layer validation is done by back-end of PHP. The PHP will validate each single input of blankness and data format, then based in validation to redirect to pages. The failure in PHP validation will result to redirect back to register page, and then in case of the successful validation, it will communicate with MySQL (AWS) database to do further validate.

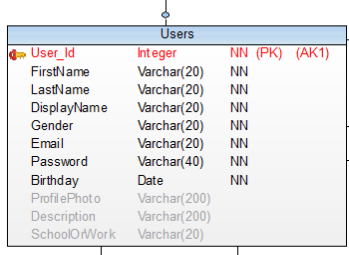
The last layer validation is done by PHP and our database. The PHP will build connection and try to insert new user information, but what if a registered user already exists in our database (same Email address as login username)? Then the SQL validation will come to function. The SQL will return a false value to PHP which tells that database insert cannot be done, so PHP can print out error and user can go back to register page to choose a different E-mail address as user account.



Validation message for duplicate user account (PHP and SQL)

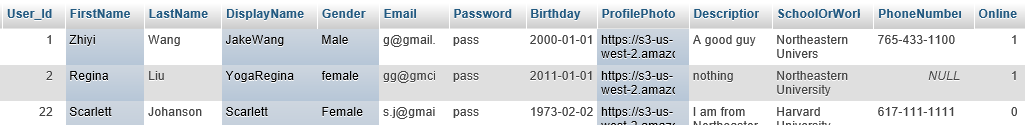
**Database design:**

When we started to design our database, we wanted to have all the attributes just like current main stream social websites, so the table includes all the common user information. The Email is the attribute for user to login, so it is set to be unique.



Initial Prototype of ‘Users’ table

And at the mid-way of development, the table has been added ‘Online’ attribute, to store the status if this user already logged in or not, which is for displaying the online status in other pages.

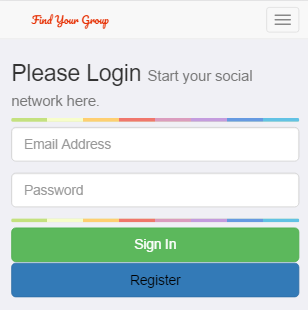
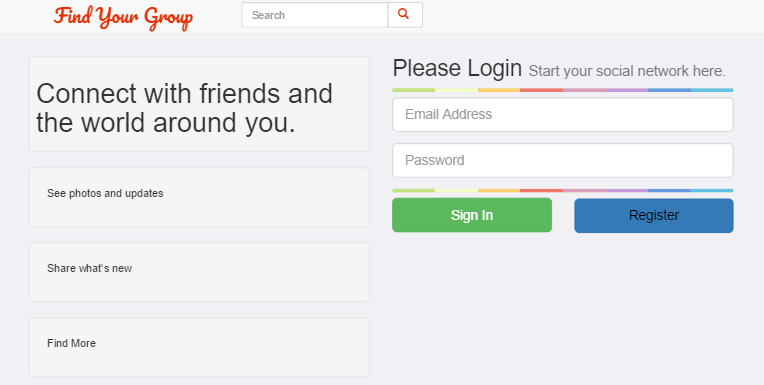


Final ‘Users’ table design with ‘Online’ status attribute

### Login

**Design:**

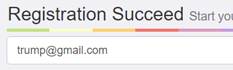
Login page is where your start to log into your account and have a glance at your own user home page, which implied a fully responsive, dynamic design for different screen sizes of devices, with both front and back end validations.

Phone Display Desktop/Tablet Display

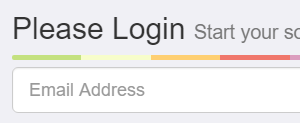
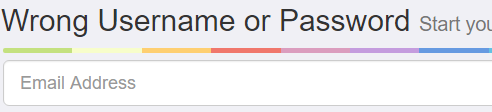
**Log-in Status:**

Above username input field, a status of current login state is showing and this login status will change based on the current state of user. When user finished the registration, and redirected from register page to login page, the status will be ‘Registration Succeed’ to indicate that your account is registered and you are good to go, with saved login username, which is email address here, you can just input password and login.



Redirect from Register page: Saved login username and status of Successful Registration

When a user open the website from Findyougroups.com, the status will be showing ‘Please Login’ for general states; and if user typed wrong username or password, the PHP and database will catch it and return a new status of ‘Wrong Username or Password’ for user.

Direct open login page Info for failed validation of username and password combination

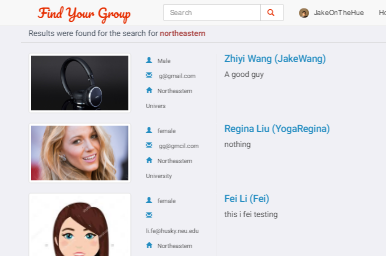
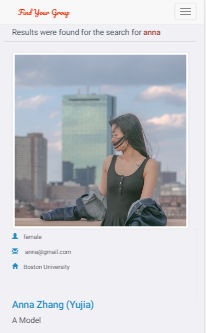
### Search (On Navigation Bar)

The search input field on navigation bar allow guest or logged in user to search other registered users, by information of name, display name, and the school or company of certain users.



Search: find people in your group.

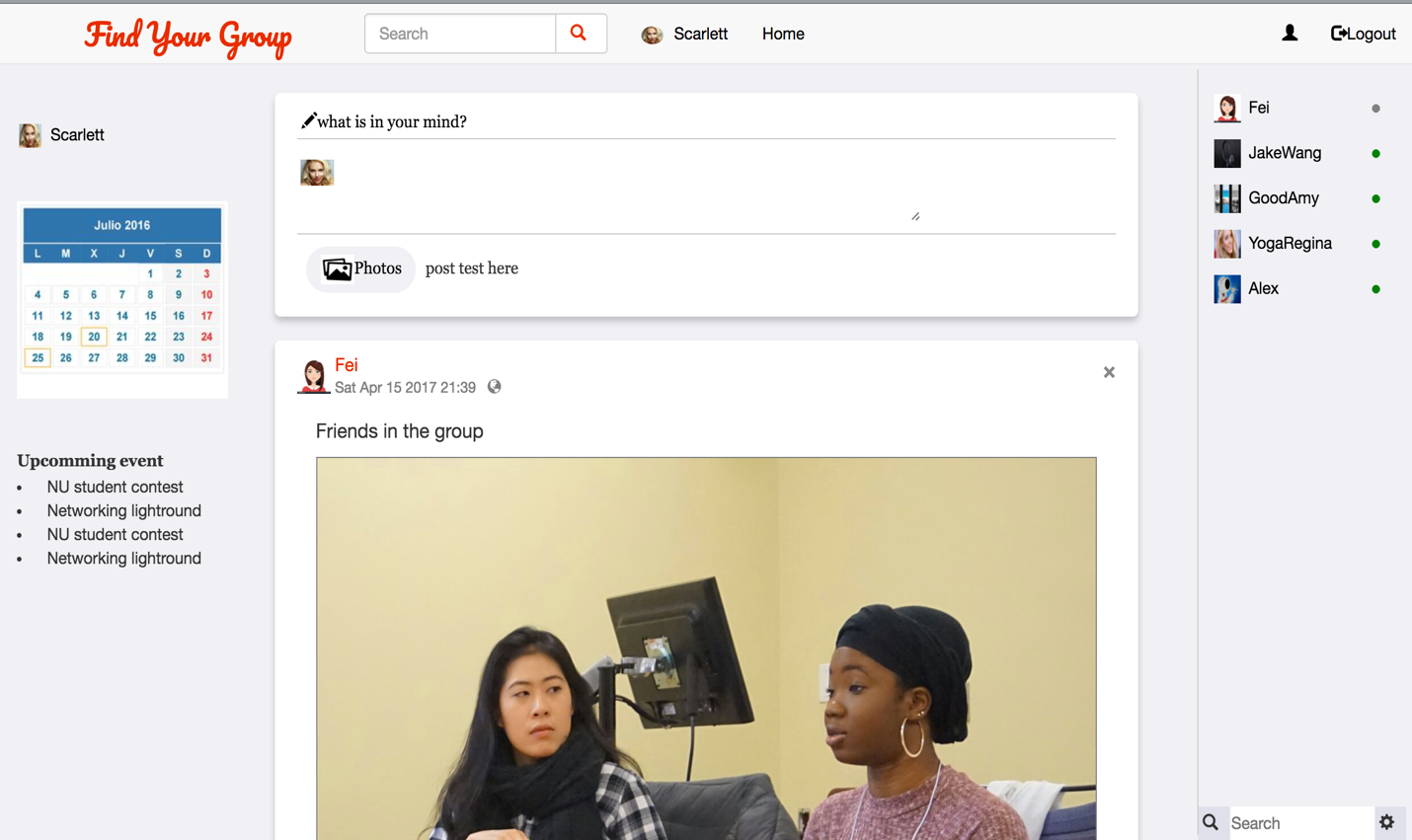
The search page is also applied with a fully responsive, dynamic design for different screen sizes of devices, with both front and back end validations.

Search by University attending Search by name

### User Home

* Navgation bar: provide navigation links to different page: personal page, friend’s personal page, edit profile.
* Friendlist right sidebar:
  + list friends and online status
  + Provide search bar, using javascript to filter
* Publish new post:
  + Use contentEditable div to provide a extendable input space
  + Upload image to the post
  + Provide preview after image selected
  + Display the post right after 'submit', using jquery
  + Save publised post to server without refresh the whole page, using Ajax
* Populate friends post:
  + Retrieve friends' posts to home page on time descending order
  + Dynamic reload when srcoll, using Jquery and Ajax
  + Provide the cross button to delete certain post from home page



### User Personal Page

This page will show up after user click on the personal page icon at navbar. It is the next step after User Home. All the information related to the user is on this page.

#### C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCacheContent.Word\friend_page.pngStructure

Here is the Friend Page website structure. Specifically, the Section Tabs navigate Sections area to display Timeline Here is the Friend Page website structure. Specifically, the Section Tabs navigate Sections area to display Timeline, About, Photos and Friends sections.

e, About, Photos and Friends sections.

#### Tab Content

In the “Section” area of the structure picture. There are four sections, “Timeline”, “About”, “Friends” and “Photos”.

In “Timeline” Section, which are the posts of the users, and structure of each post is as followed,

C:\Users\Alex\AppData\Local\Microsoft\Windows\INetCacheContent.Word\post.png

Here is the panel structure for each user, particularly, we implemented AJAX to partially refresh only the footer part, which I mean is the comment part of the post. Similarly, users can delete a post without refreshing the whole page, but only the post div section.

“About” Section contains the basic information, such as school, work, phone number, of the users. And also, here is a portal for users to update their basic information.

“Friends” Section, just the same with the name, which contains the list of the users` friends. Users can “unfriend” their friends in this section.

“Photos” Section is collecting all the photos that users have uploaded through their posts. There is a subpage called “Photo Wall” to display the photos in a fancier way.

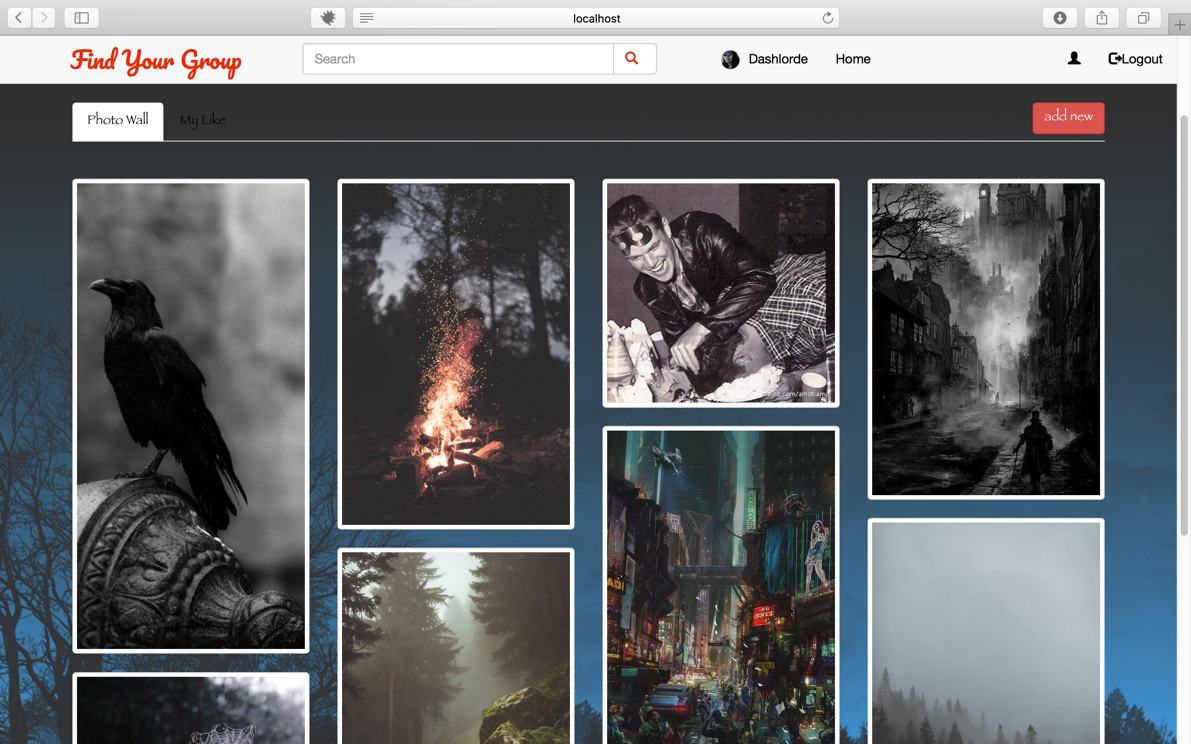
### Friend Page

The whole structures of Friend Page and the subpages are similar to User Page. The differences are as follows:

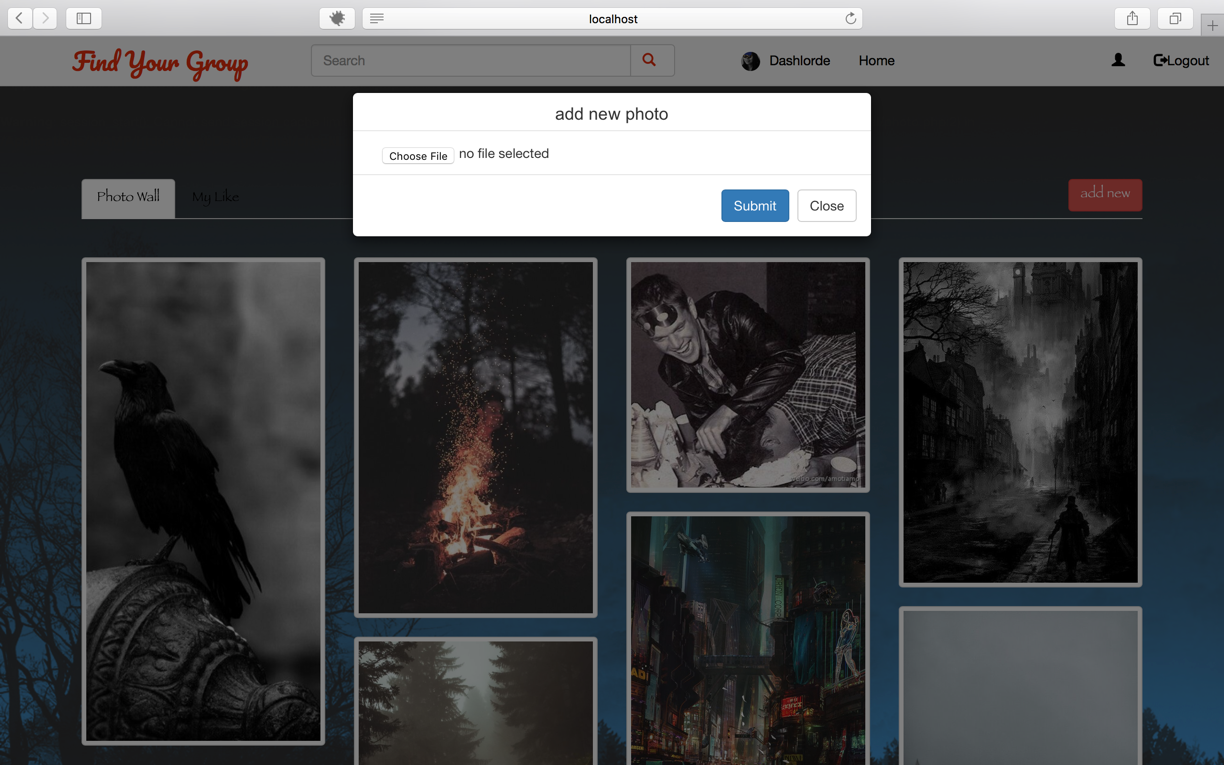
* There is “Add to Friend” button on a friend page`s cover.
* A user can leave a comment to a friend post, but cannot delete a post.
* A user can add or delete a friend through his or her friend`s “Friends” Section.
* A user cannot update friends` basic information.

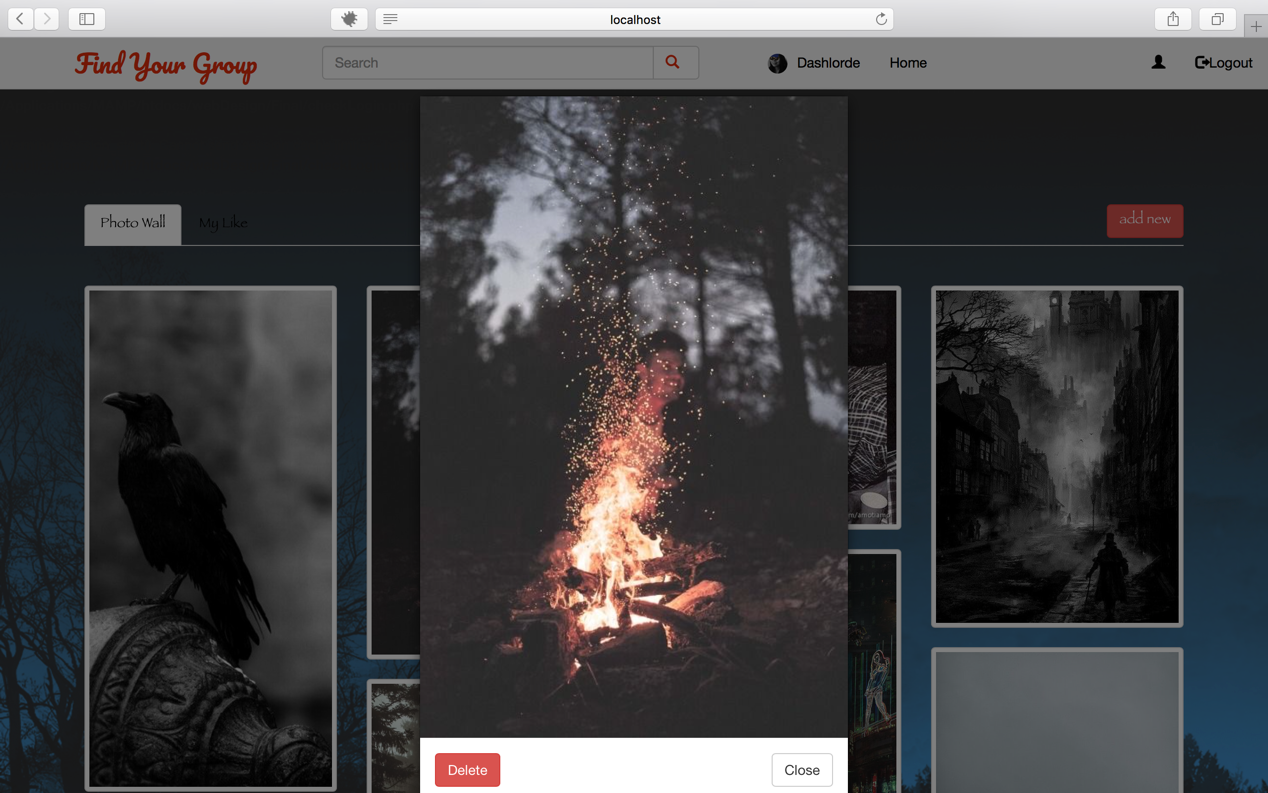
### Photo Wall:

In Photo Wall page, a jQuery plugin named masonry is implemented to create a Pinterest-like layout.

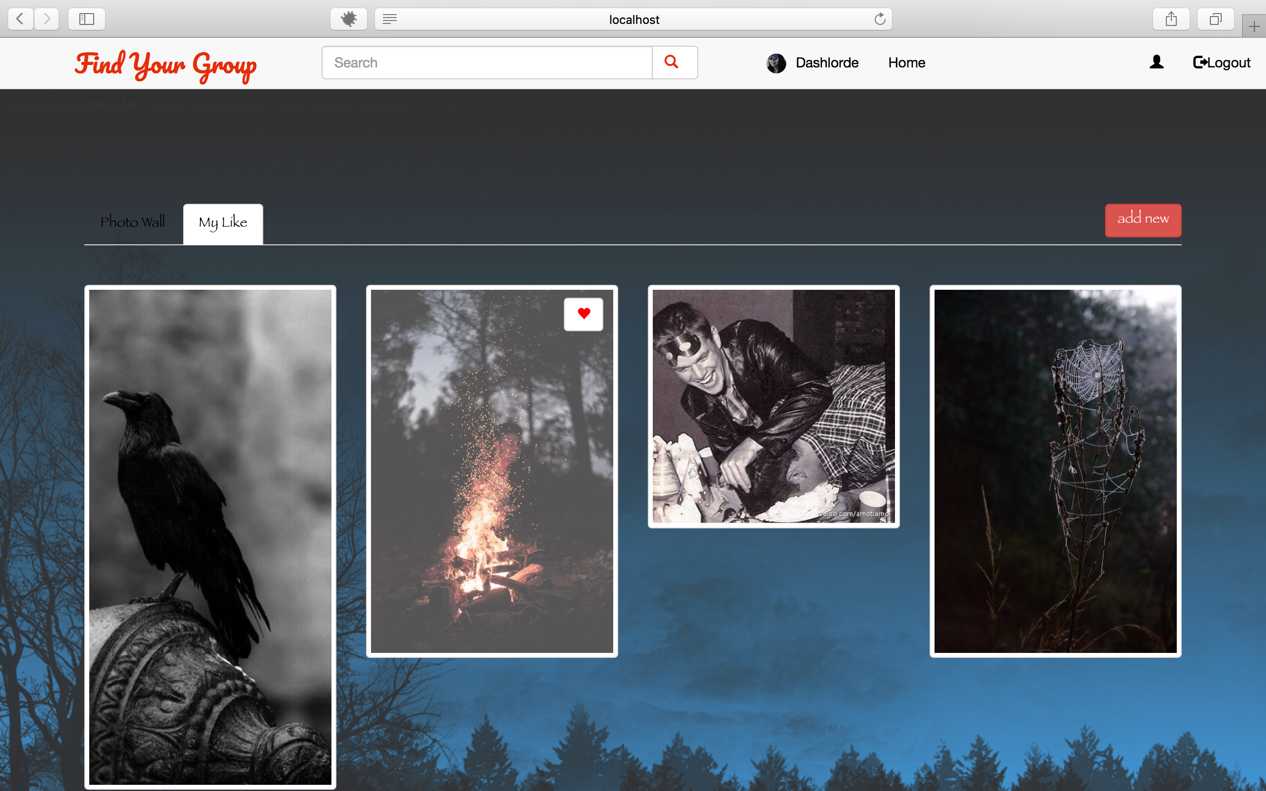


User can upload photo directly in Photo Wall, focus on one of the pictures and delete picture.



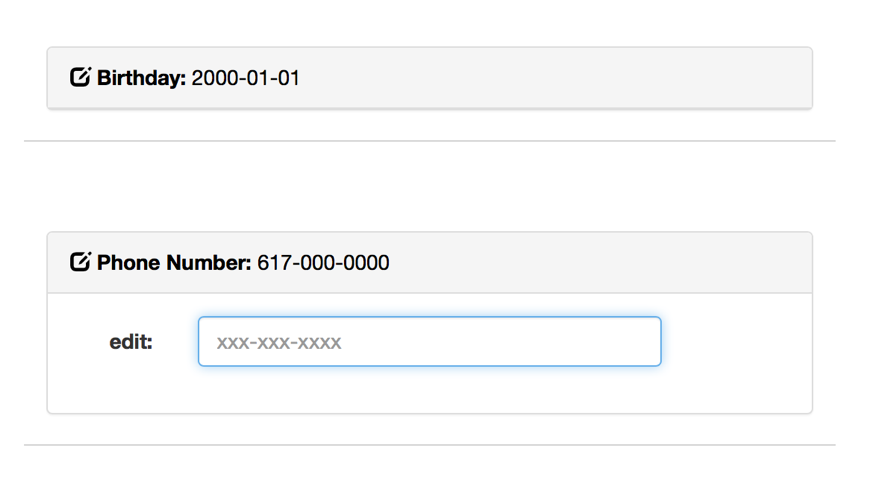


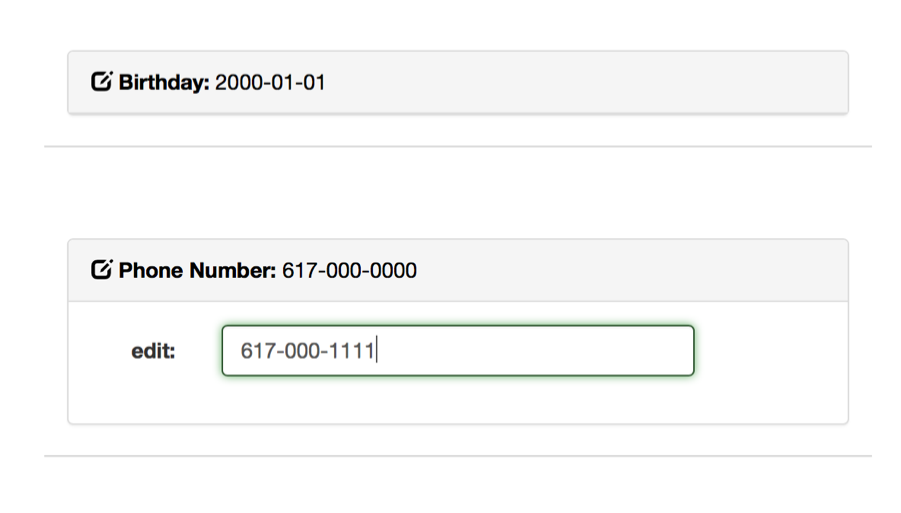
Moreover, user can add photo to like list, and all liked photo will be displayed in My Like tab content. The color of heart icon will change if user add the picture to like list.

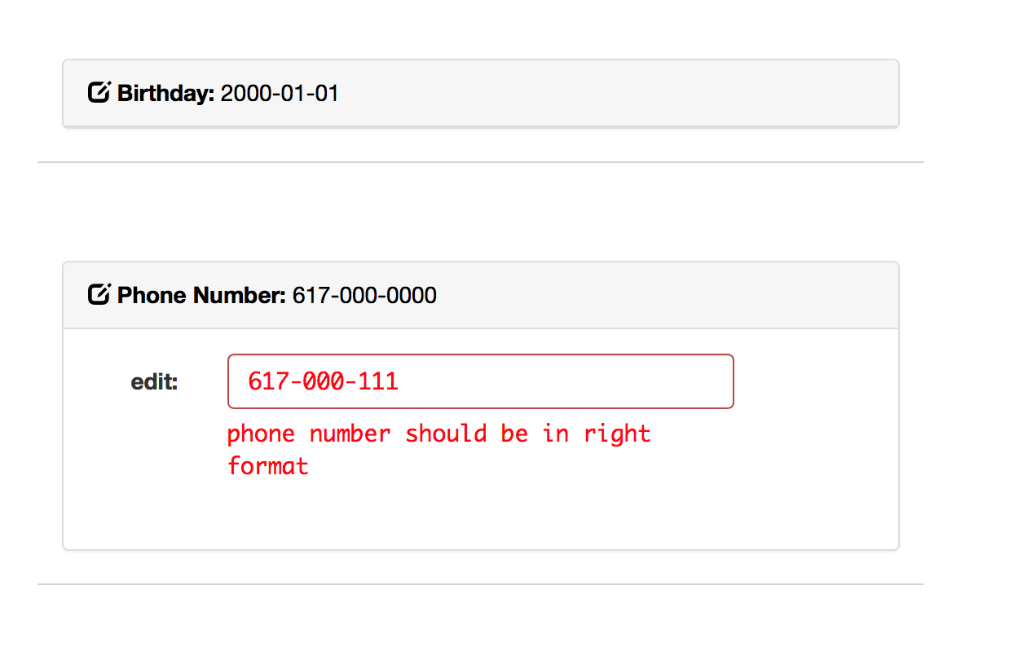


### Profile:

In profile page, a JQuery validation plugin is implemented to validate user’s input. If user input wrong format, error message will be shown immediately. In addition, PHP validation is also configured a more security environment.







### Database and picture stored configuration

To conveniently synchronized database and pictures in team work. We use Amazon Web Services to create remote database and a bucket for user picture storage. Therefore, this project cannot run directly. Some files need to edit in order to run with local database and local picture storage, or RDS and S3.

The configuration detail is written in readme file in the project root folder.