

# CS6314.501 - Team1 - Fun Share Social Media Website

Xiaojing Zhang, Yang Song, Jing Zhou, Chunhe Ni

## 1 Website Name

The name of the website is Fun Share. We want our users can share their fun on the website.

## 2 Team Members

Xiaojing Zhang	xxz141130
Yang Song	yxs167230
Jing Zhou	jxz160330
Chunhe Ni	cxn170001

## 3 Project Description

The website we built is a social media website.

- Users can register their accounts and login to our website and log out.
- Users can do a lot on the website, like: post pictures, videos they want to share.
- Users can delete the posts they posted.
- Follow people they are interested in.
- Users can set they posts private, public and friends only.
- Users can click like for the posts they like.
- Comment on our people's posts with text, pictures and videos.
- We set notification for the comments, when user get new comments, they will get the notification first, and choose if the comments are allowed to show after their posts.
- They can send messages to the other users and receive and reply message to other users.
- Users can search other users and other user's posts.
- Search results can be filtered by user's gender.
- Admin can delete users' posts and account.
- When user logged on for two hours, our website will notice them, and pop up kind advice.

## 4 Database Design

Our database is funshare and it has 7 tables:

1. User table: it stores all the user information, which includes user\_id, username, email,

salted\_password, first\_name, last\_name, profile\_photo (which is the path of the profile photo) , private (indicates the whether the account is public), only\_friend (indicates whether the account is only friend visible), isAdmin(whether the user is administrator), gender, date\_created and date\_updated.

2. Photo table: it stores all the photo/video information of a post, which includes photo\_id, user\_id (who post this photo/video), caption (name of the photo/video), image\_path, likes (how many likes the photo/video got), date\_created and date\_updated.
3. Posts table: it stores the text content about a post, which includes post\_id, user\_id (who made this post), photo\_id (the corresponding photo\_id of the post), post\_content and date\_created.
4. Comment table: it includes all the text comment. It has primary key – comment\_id, foreign keys – user\_id (who made this comment) and photo\_id (comment on which photo). It also has 3 other attributes – comment is used to store the content of the comment, authorize is used to indicate whether the comment is allowed to be posted, and the date\_created indicates the date was created.
5. Photo\_comment table: it stores comment which is a photo or video itself. It has attributes: comment\_photo\_id, photo\_id (the post it commented), comment\_id (the corresponding text comment id), photo\_path and date\_created.
6. Messages table: it stores all the message information between users. It has attributes: message\_id, message\_body (message content), sender, receiver, isread(whether the message has been read) and date\_created.
7. Following table: it stores the following relationship between users. It has attributes: follow\_id, followee (the user being followed) and follower (the user follows other user).

## 5 Languages/Frameworks

Front End:

HTML, Bootstrap, Javascript, JQuery, Ajax

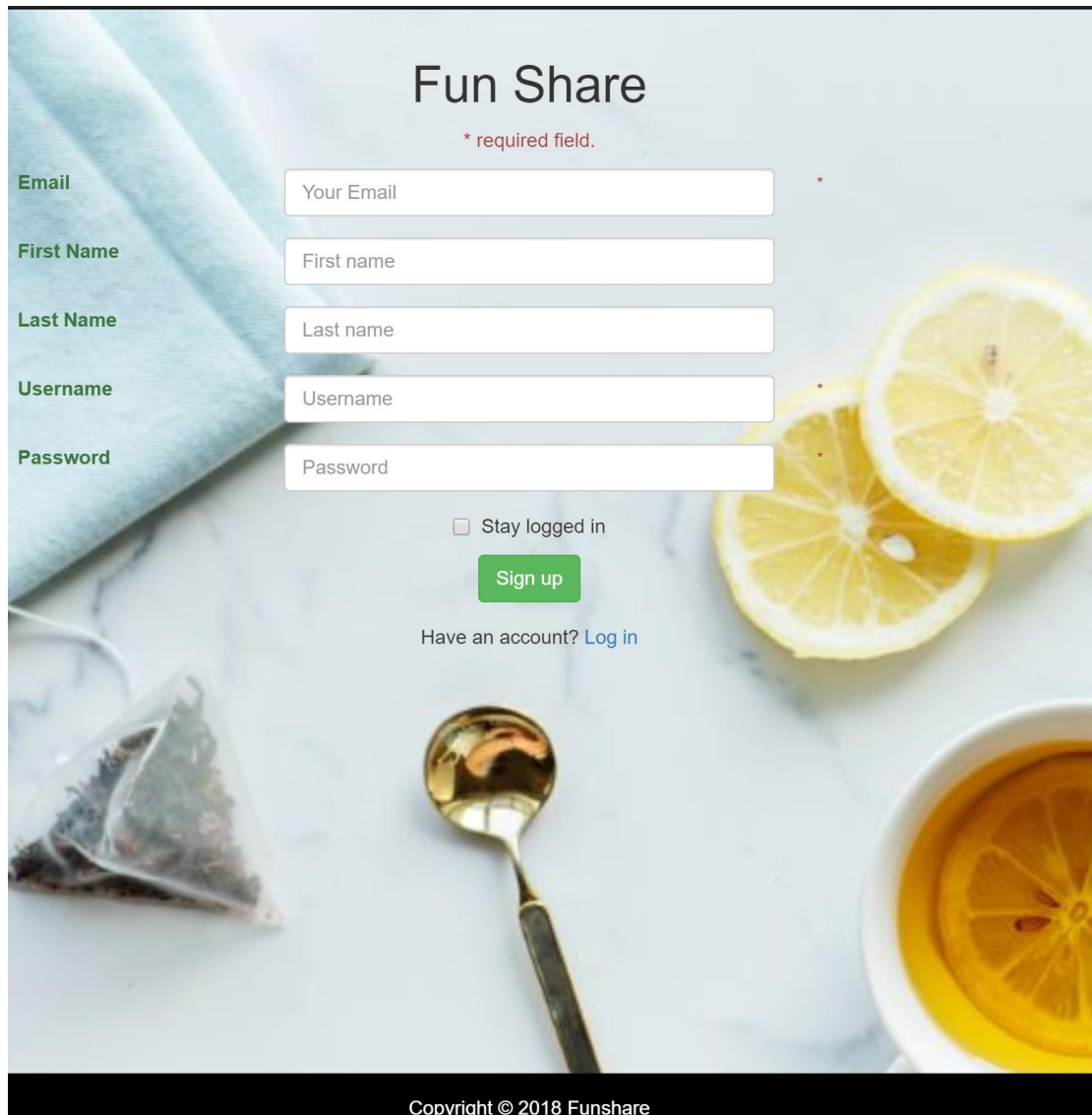
Back End:

PHP

Database:

mySQL

## 6 Screenshots



The image shows a registration form for 'Fun Share' overlaid on a lifestyle background. The background features a light blue cloth, lemon slices, a tea bag, a spoon, and a cup of tea. The form is centered and includes the following elements:

### Fun Share

\* required field.

**Email**

**First Name**

**Last Name**

**Username**

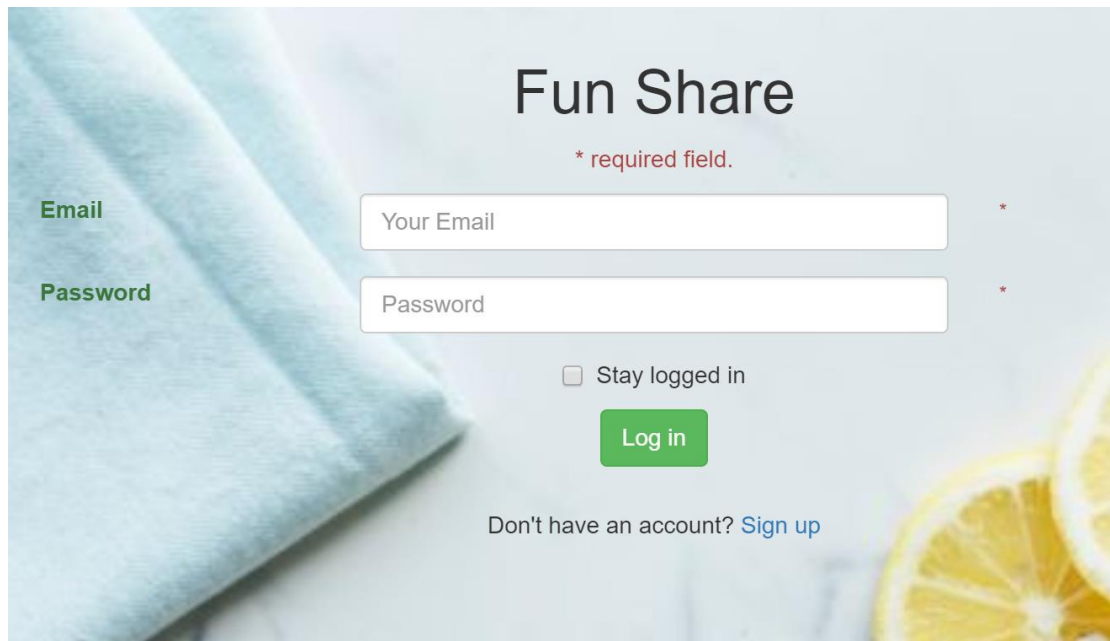
**Password**

☐ Stay logged in

Have an account? [Log in](#)

Copyright © 2018 Funshare

Figure 1 Register



The login page for 'Fun Share' features a light blue background with a close-up of a folded cloth on the left and a lemon slice on the right. The title 'Fun Share' is centered at the top. Below it, a red asterisk and the text '\* required field.' are displayed. The 'Email' label is in green, followed by a text input field containing 'Your Email' and a red asterisk to its right. The 'Password' label is also in green, followed by a text input field containing 'Password' and a red asterisk to its right. Below the password field is a checkbox labeled 'Stay logged in'. A green 'Log in' button is centered below the checkbox. At the bottom, the text 'Don't have an account? Sign up' is displayed, with 'Sign up' as a blue link.

# Fun Share

\* required field.

Email

Password

☐ Stay logged in

[Log in](#)

Don't have an account? [Sign up](#)

Figure 2 Log In

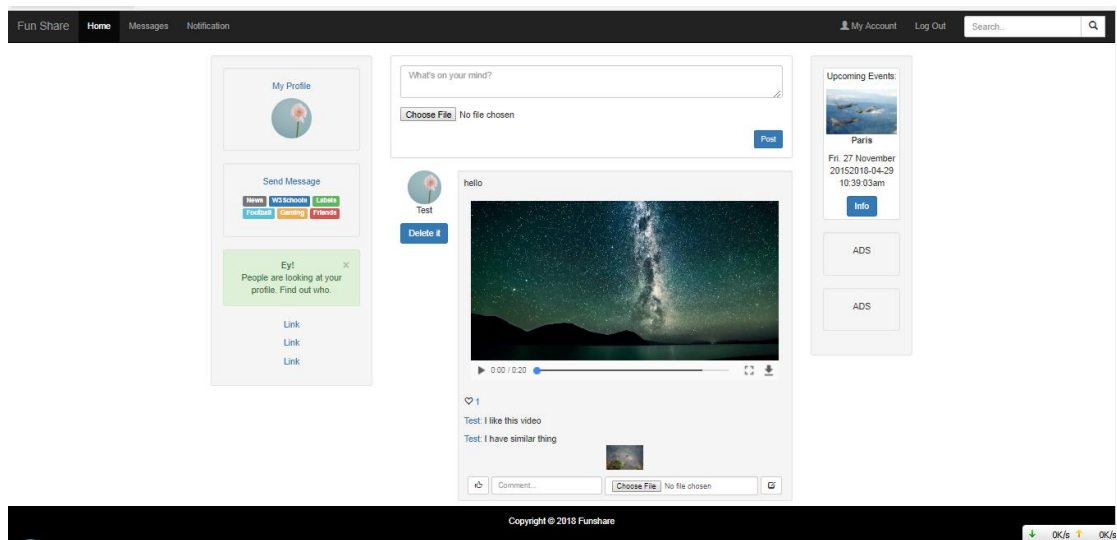


Figure 3 Profile Viewed by Admin

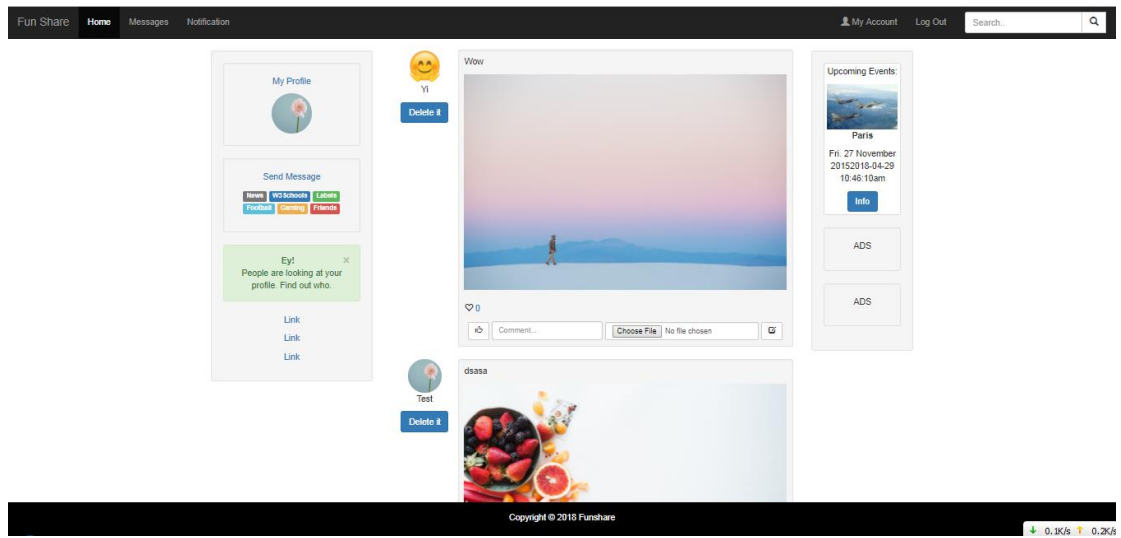


Figure 4 Admin Homepage



Figure 5 Comment Notification

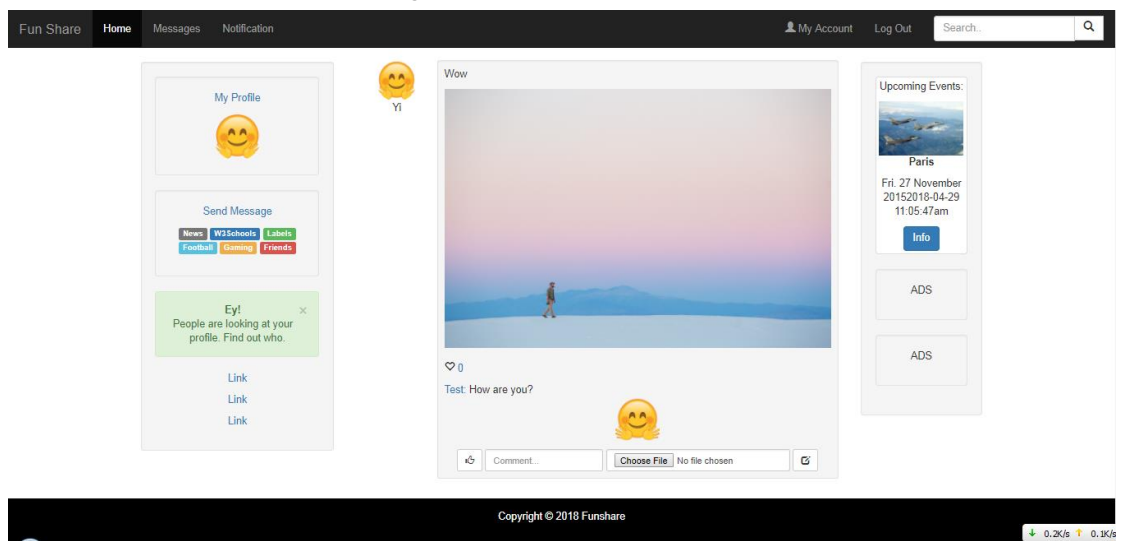


Figure 6 Non-admin User Homepage

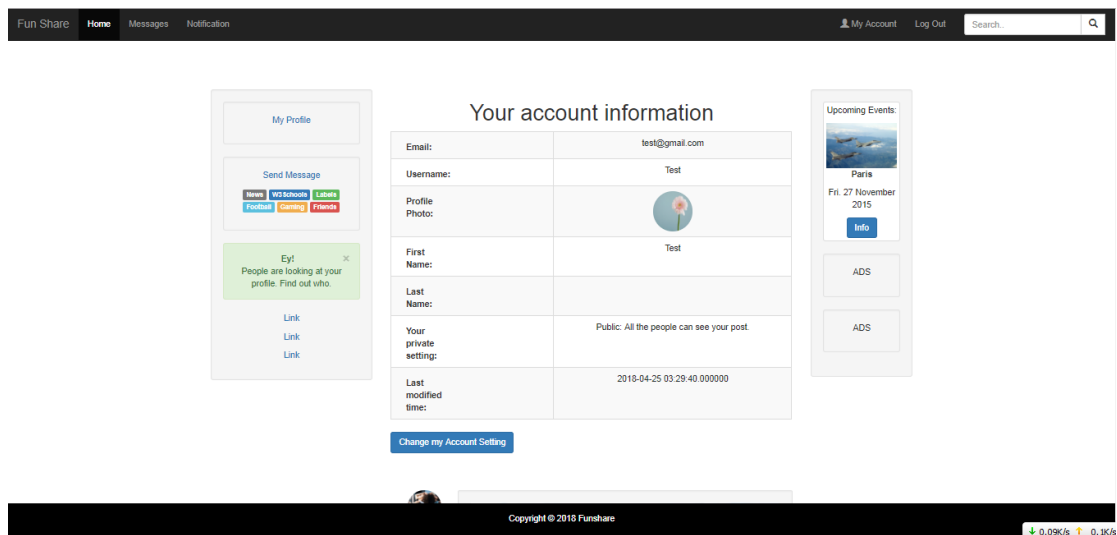


Figure 7 My Account Page

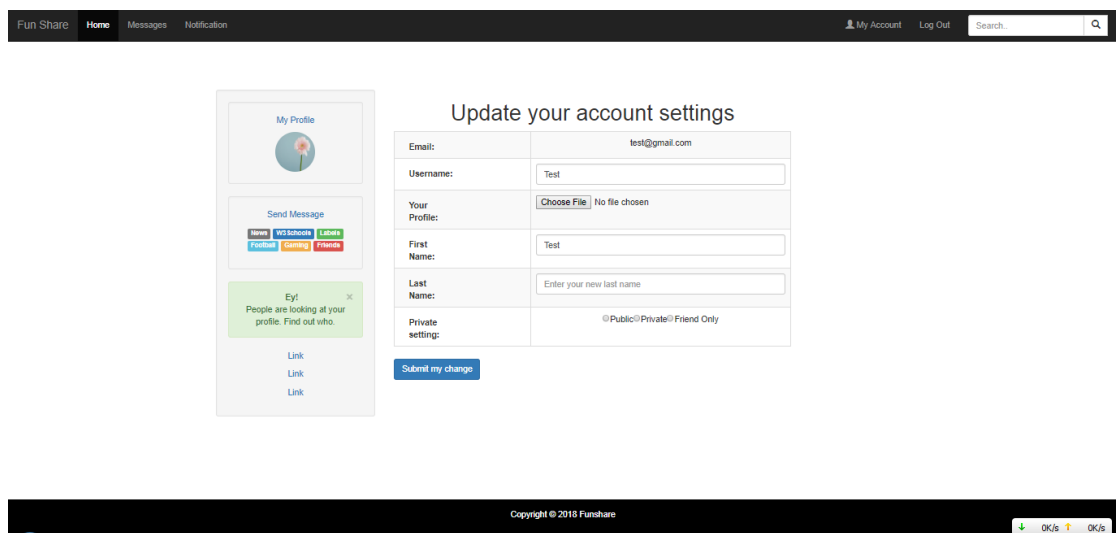


Figure 8 Modify Account Page

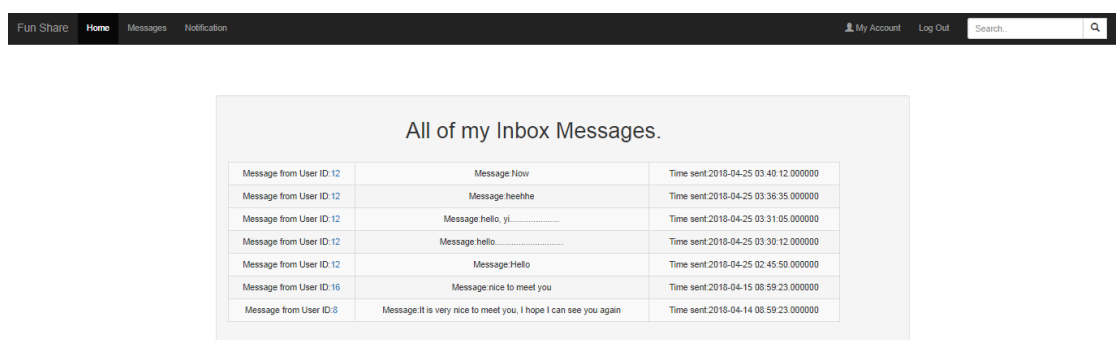


Figure 9 Message Box Page

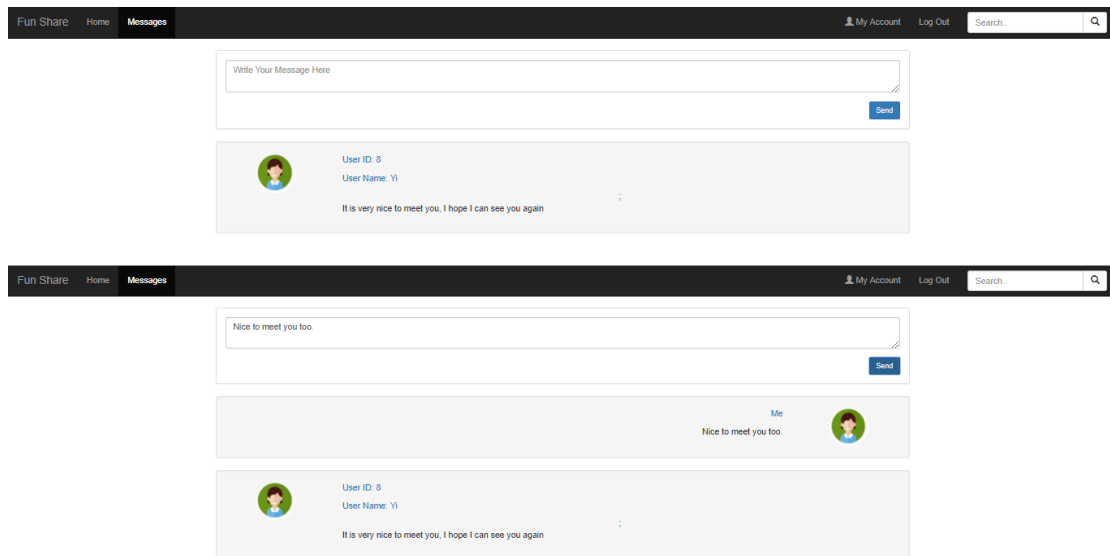


Figure 10 Message Page

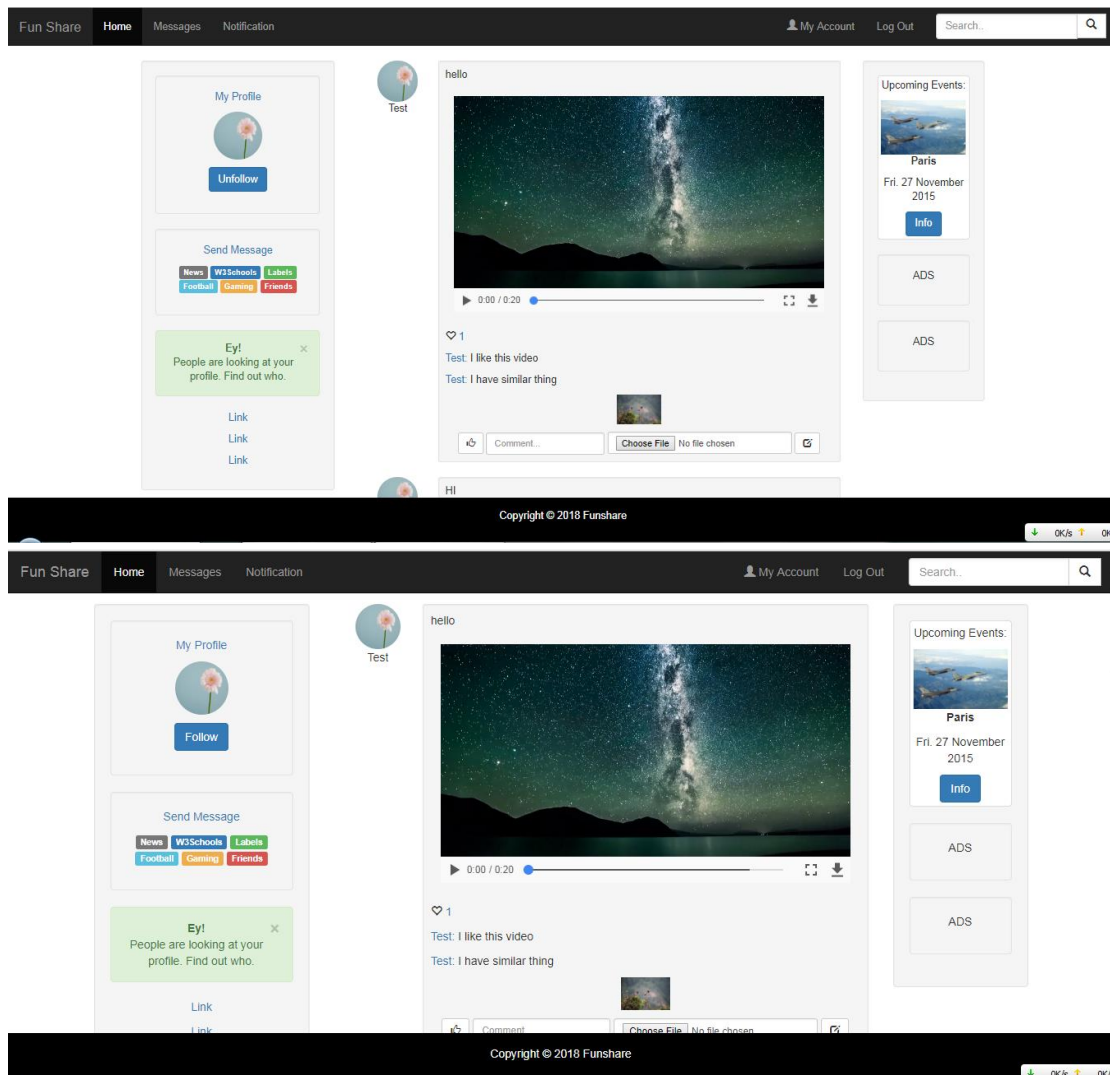


Figure 11 Other User's Profile Page with Follow/Unfollow Button

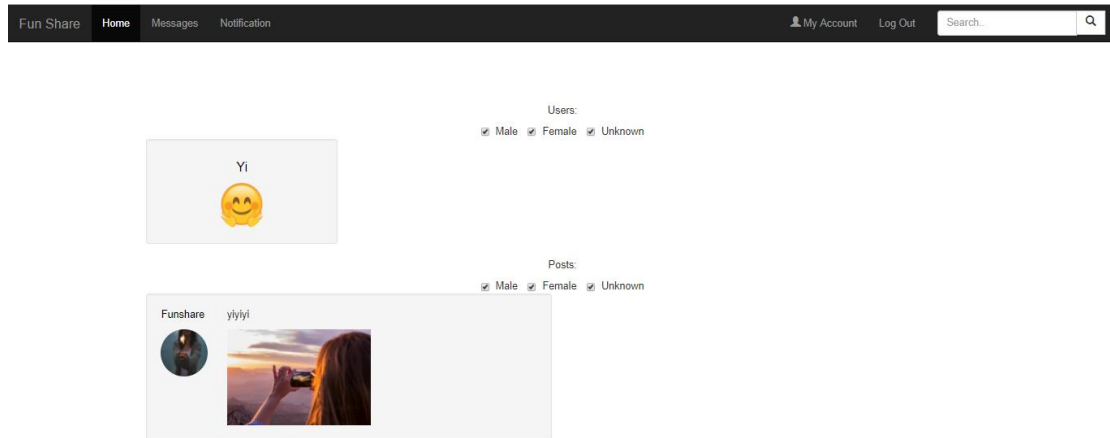


Figure 12 Search Result with Filter

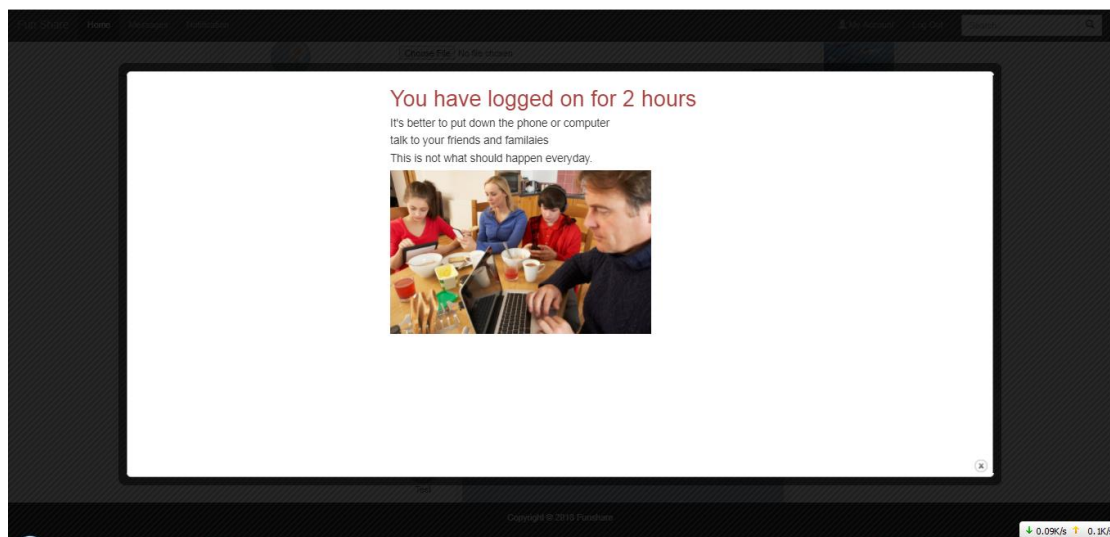


Figure 13 Reminder Feature

## 7 Work Division

### 7.1 Xiaojing Zhang

Designed and implemented the frontend and backend signup and login pages. Implemented likes feature and comment feature, allowing people comment text, picture and videos. Implemented follow feature, allowing people follow other people. Implemented privacy setting feature such that user's account can be private, public or only friend visible. Combined and perfected the frontend and backend of homepage and profile pages, fixed bugs. Designed and finalized database schema. Extracted footer, header, connected DB etc. files to make it more modular.



## 7.2 Yang Song

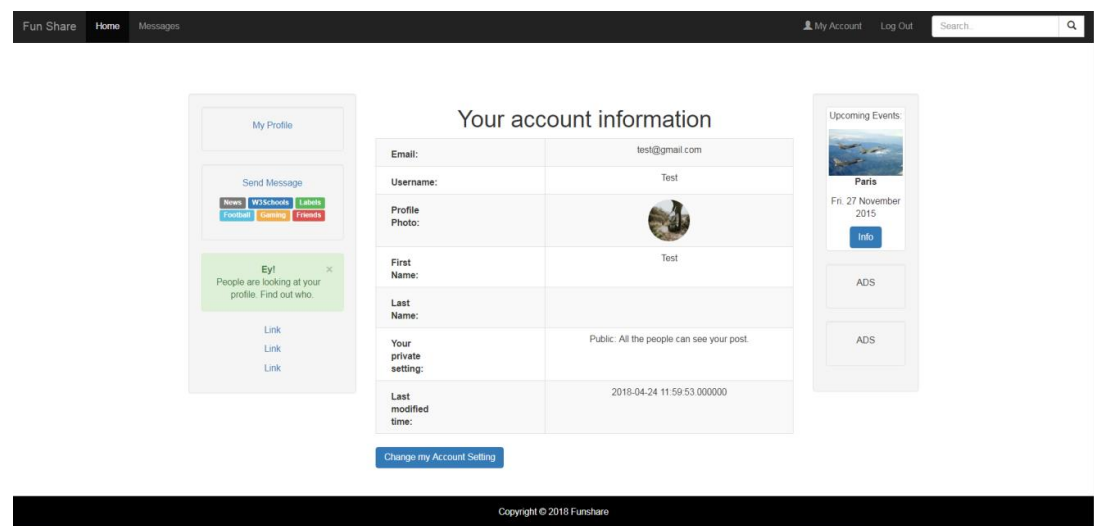
Designed the post part, which completed users can post different kind of content, include text, pictures and videos and show the posts in their profile and homepage by timeline. Designed the admin part, make the admin can access user's accounts, delete users posts and accounts. Designed the popup window part which is a part of the new features of us project. Set the time when a window popup to warn the user has spend a lot of time and give them kind advice.

## 7.3 Jing Zhou

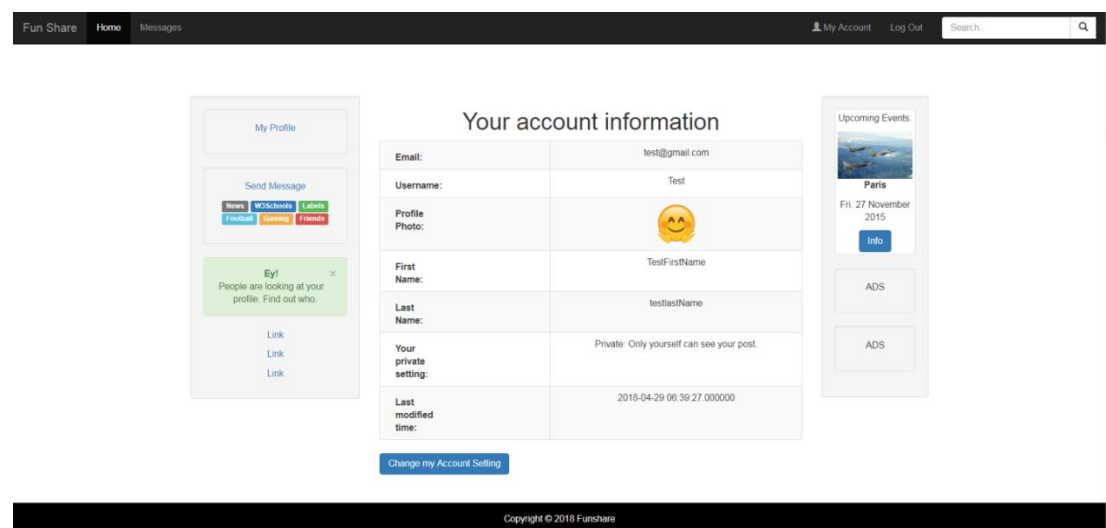
Designed and implementation account part and message part of the website.

For the account page, user can change their username, first name, last name, private setting and update their profile photo. In this part, Xiaojing helped me to finish the update profile photo function.

Before:



After:



For the message part, user can get all the inbox message by click the messages button in the header bar. They can reply the message by click the user id number.

[Fun Share](#) [Home](#) [Messages](#) [My Account](#) [Log Out](#)


### All of my Inbox Messages.

Message from User ID:12	Message:hellooooooooooooo	Time sent:2018-04-24 11:56:40.000000
Message from User ID:12	Message:hello, I am Jing	Time sent:2018-04-24 11:56:02.000000
Message from User ID:19	Message:11111111111111111111111111111111	Time sent:2018-04-24 08:14:07.000000
Message from User ID:19	Message:hello, I am Jing	Time sent:2018-04-24 08:01:12.000000
Message from User ID:12	Message:hello	Time sent:2018-04-24 07:52:30.000000
Message from User ID:18	Message:hello, I am weiwei	Time sent:2018-04-24 07:48:58.000000
Message from User ID:16	Message:nice to meet you	Time sent:2018-04-15 08:59:23.000000
Message from User ID:8	Message:It is very nice to meet you, I hope I can see you again	Time sent:2018-04-14 08:59:23.000000

People can also click the send message link in the left bar. To send message to others. In the pictures I show below, I created a new account named hello. He send a message to the test account.

[Fun Share](#) [Home](#) [Messages](#) [My Account](#) [Log Out](#)


This message is from hello



Send

Me

This message is from hello



Then I logged the “Hello” account out, and log in the test account. Click the messages button and get the info below.

# All of my Inbox Messages.


Message from User ID:20	Message:This message is from hello	Time sent:2018-04-29 06:52:11.000000
Message from User ID:12	Message:hellooooooooooooo	Time sent:2018-04-24 11:56:40.000000
Message from User ID:12	Message:hello, I am Jing	Time sent:2018-04-24 11:56:02.000000
Message from User ID:19	Message:11111111111111111111111111111111	Time sent:2018-04-24 08:14:07.000000
Message from User ID:19	Message:hello, I am Jing	Time sent:2018-04-24 08:01:12.000000
Message from User ID:12	Message:hello	Time sent:2018-04-24 07:52:30.000000
Message from User ID:18	Message:hello, I am weiwei	Time sent:2018-04-24 07:48:58.000000
Message from User ID:16	Message:nice to meet you	Time sent:2018-04-15 08:59:23.000000
Message from User ID:8	Message:It is very nice to meet you, I hope I can see you again	Time sent:2018-04-14 08:59:23.000000

Then I clicked the user id 20.


Fun Share

Home

Messages


 My Account

Log Out



Write Your Message Here

Send



User ID: 20  
User Name: hello

This message is from hello

I can reply the message through this page.

In this message part, Chunhe helped me in making the message update automatically by using ajax.

## 7.4 Chunhe Ni

Designed and Implemented the user interface of the website, the notification page and the search page.

For user interface, Bootstrap is mainly used to modify the page structure. With the help of the grid in Bootstrap, it is easy to divide the page content by using the css class like “row” and

“col-sm-xx”.

For notification part, each comment has an attribute called “authorize” to identify the authorize status. Initially, the authorize attribute is set to 0 which means the comment has not been handled by the user. If the user clicks the “show” button, the authorize will be set to 1 which means it can be shown under the user’s post. If “delete” is clicked, the authorize will be set to 2 and cannot be shown under the post.

For search page, the search function is implemented by using the “LIKE” keyword in mySQL query. For example, using “username LIKE %tom%” in the query, mySQL will return all the user whose username contains “tom”. The search page also has a filter implemented with jQuery. Each result returned by database will be display in a <div> tag on the page. When the user clicks the checkbox, jQuery is used to select the related <div> tags and show or hide them.