**CST8257 Web Application Development**

Project

# Objective

1. Apply knowledge learnt in this course to create a social media network website for picture album management and sharing.

# Due Date

See Canvas post for the due of this project. To earn 20 points, you are required:

1. Complete this project in a group of 2 – 3 students as required.
2. Submit your project work to the Canvas before the due date. Only one submission is required for each group.
3. Demonstrate your project work during the lab session before the due date.

# Requirements

This is the comprehensive project of this course. Compared with the other labs, the amount of workload evolved in this project is significant. Your final deliverables should be at “**commercial-grade**”. To be specific:

1. The final pages should be well styled like being developed by a professional web developer.
2. The website is secure; this is, it is protected from password stealing, SQL injection and XSS attacks.

# Database

1. Create the following tables in the database **CST8257**:

**User**

UserId varchar(16) NOT NULL PRIMARY KEY,

Name varchar(256) NOT NULL

Phone varchar(16)

Password varchar(256)

**Accessibility**

Accessibility\_Code varchar(16) NOT NULL,

Description varchar(127) NOT NULL,

**Album**

Album\_Id int(11) PRIMARY KEY AUTO\_INCREMENT,

Title varchar(256) NOT NULL,

Description varchar(3000),

Date\_Updated Date NOT NULL,

Owner\_Id varchar(16) NOT NULL,

Accessibility\_Code varchar(16) NOT NULL,

FOREIGN KEY (Owner\_Id) REFERENCES **Student** (StudentId),

FOREIGN KEY (Accessibility\_Code) REFERENCES **Accessibility** (Accessibility\_Code)

**Picture**

Picture\_Id, int(11) NOT NULL PRIMARY KEY AUTO\_INCREMENT

Album\_Id, int(11), NOT NULL

FileName, varchar(255) NOT NULL

Title, varchar(256) NOT NULL,

Description, varchar(3000),

Date\_Added date NOT NULL,

FOREIGN KEY (Album\_Id) REFERENCES **Album** (Album\_Id)

**Comment**

Comment\_Id int(11) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Author\_Id varchar(16) NOT NULL,

Picture\_Id int(11) NOT NULL,

Comment\_Text varchar(3000) NOT NULL,

Date timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (Author\_Id) REFERENCES **Student** (StudentId),

FOREIGN KEY (Picture \_Id) REFERENCES **Picture** (Picture \_Id)

**FriendshipStatus**

Status\_Code varchar(16) NOT NULL PRIMARY KEY,

Description varchar(128) NOT NULL,

**Friendship**

Friend\_RequesterId varchar(16) NOT NULL,

Friend\_RequesteeId varchar(16) NOT NULL,

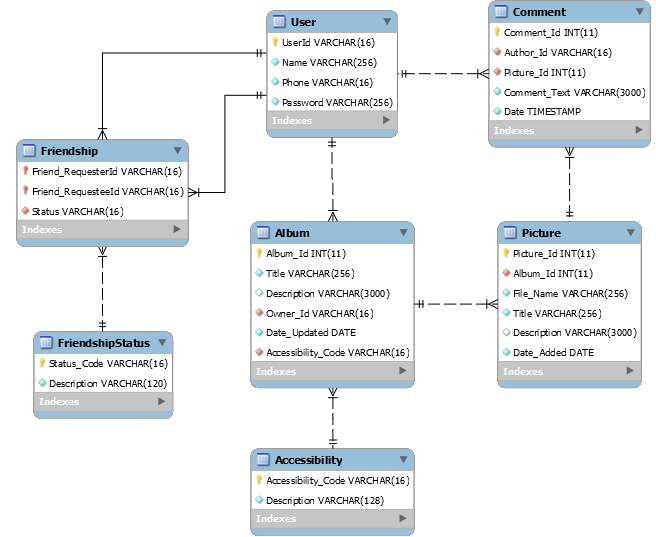
Status varchar(16) NOT NULL,

FOREIGN KEY (Friend\_RequesterId) REFERENCES **Student** (StudentId),

FOREIGN KEY (Friend\_RequesteeId) REFERENCES **Student** (StudentId),

FOREIGN KEY (Status) REFERENCES **FriendshipStatus** (Status\_Code)

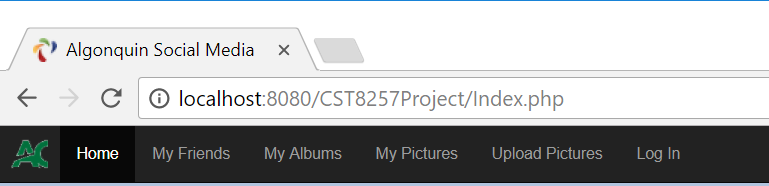
1. Run SQL statements in file “**Script for Populating Project DB tables.sql**” (downloadable from the Canvas) against database CST8257 to populate **Accessibility** table and **FriendshipStatus** table.
2. The follow is the relational data model of this website. Study this relational data model to see how you can derive entity classes from the database tables in the model.



# Website

1. **header.php** and **footer.php**

All pages on this website have the following header.



The header.php page should have the following menu items:

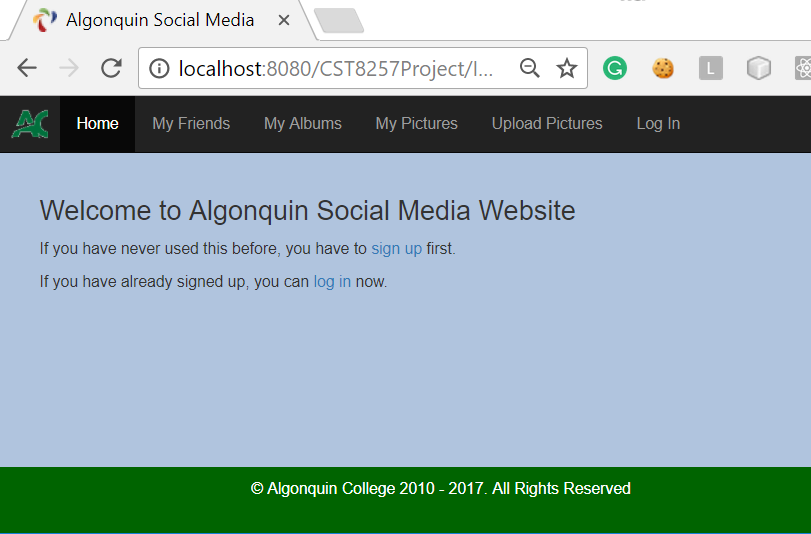
* 1. **Home** – link to Index.php page
  2. **My Friends** – link to MyFriends.php page
  3. **My Albums** – link to MyAlbums.php page
  4. **My Pictures** – link to MyPictures.php page
  5. **Upload Pictures** – link to UploadPictures.php page
  6. **Log In / Log Out** – link to Login.php if the user is not logged in yet, or Logout.php page if the user has already logged in.

All pages on this website should have the following footer:



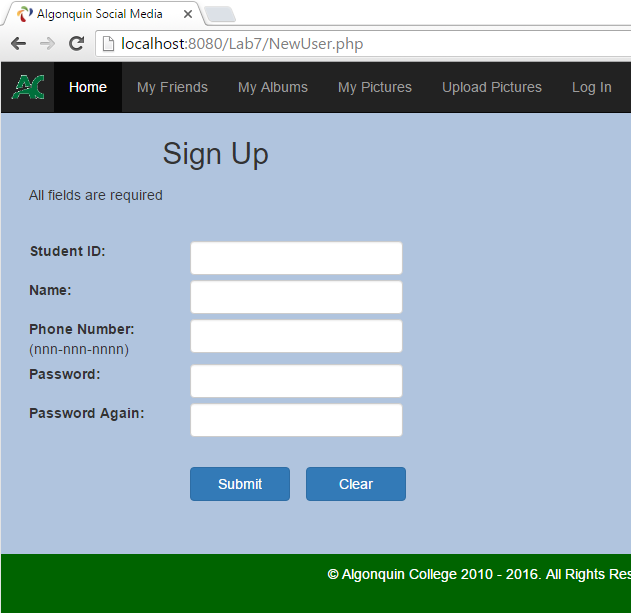
1. **Index.php**

This page is the landing (default) page of this website.



1. **NewUser.php**

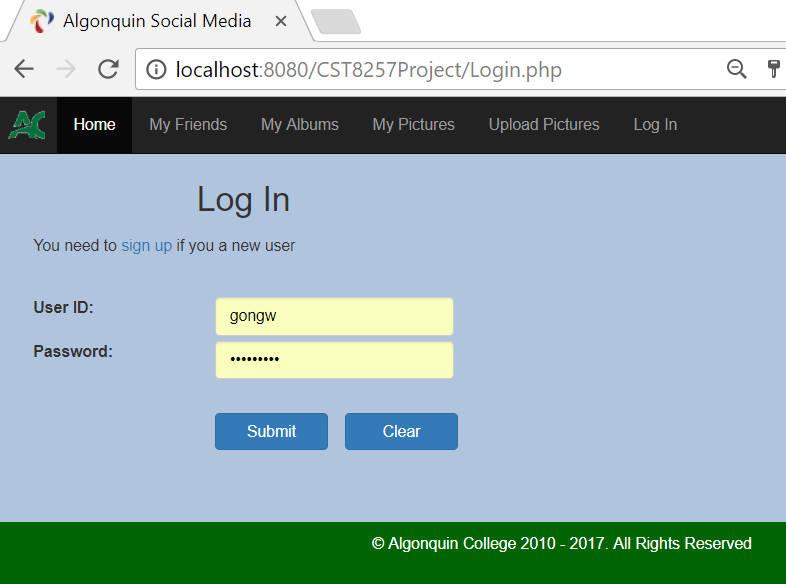
The same as in lab 7, this page is for a user to sign up with the system. The same validation rules as in lab 7 apply as well. In this lab, passwords cannot be saved in the User table as plain text. They must be encrypted.



1. **Login.php**

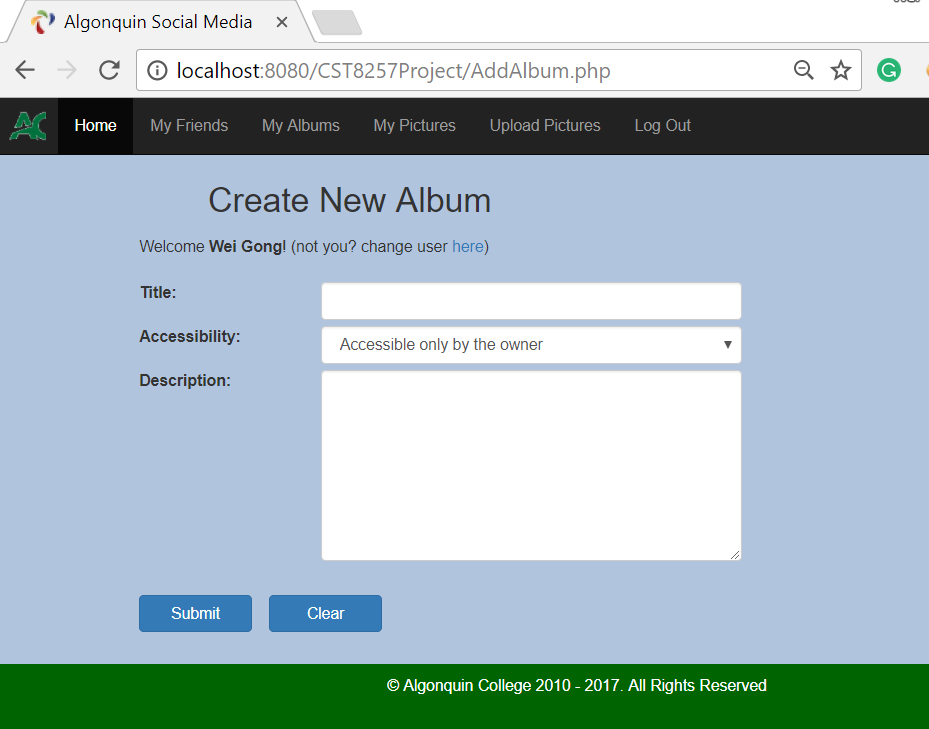
This page allows the signed-up users to log into the website with their student ID and password.

Except the Landing page (index.php), New User page (NewUser.php) and this page, all other pages of the website are protected, requiring users to be authenticated to access. If an unauthenticated user tries to access a protected page of the website, he/she will be redirected to this login page for authentication. Once successfully authenticated, the user will be redirected back to the protected page he/she was attempting to access.



1. **AddAlbum**

This page is used to create a new album for the user.



On this page, the user specifies the title of the album, select the accessibility from the dropdown list and optionally the detailed description of the album.

The possible selections of dropdown list come from CST8257 database table **Accessibility**. Currently there are two entries in the table:

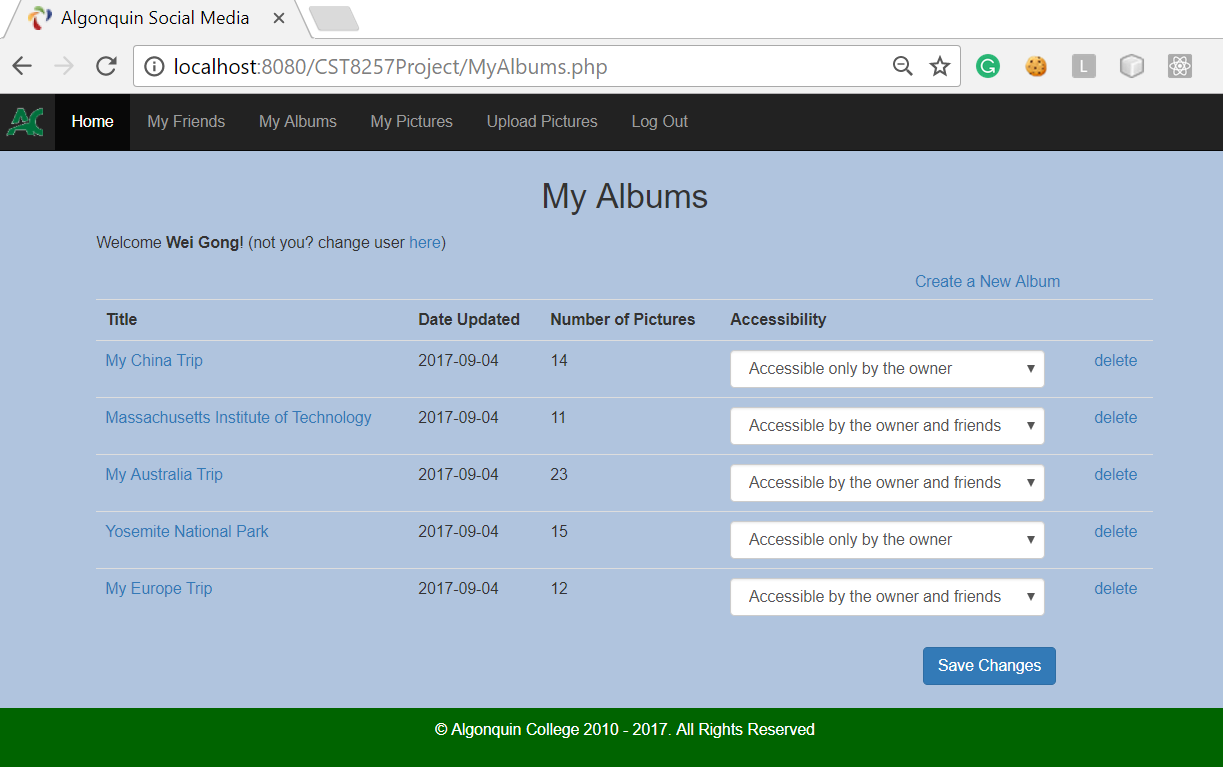
* private – The album is only accessible by the user him/her self.
* shared – The album is accessible by the user and the user’s friend.

1. **MyAlbums.php**

This page lists user’s albums. For each album, it lists the following information about the album:

* Title – The title of the album
* Date Updated – the last date when the user uploaded the picture(s).
* Number of pictures – The number of pictures the album contains
* Accessibility – the album’s accessibility. It is shown as the selection of the dropdown list. The available selections in the dropdown list come from the database table Accessibility

The user can change the of albums’ accessibility by selecting the desired accessibility from the dropdown list for each album and click the **Save Change** button.

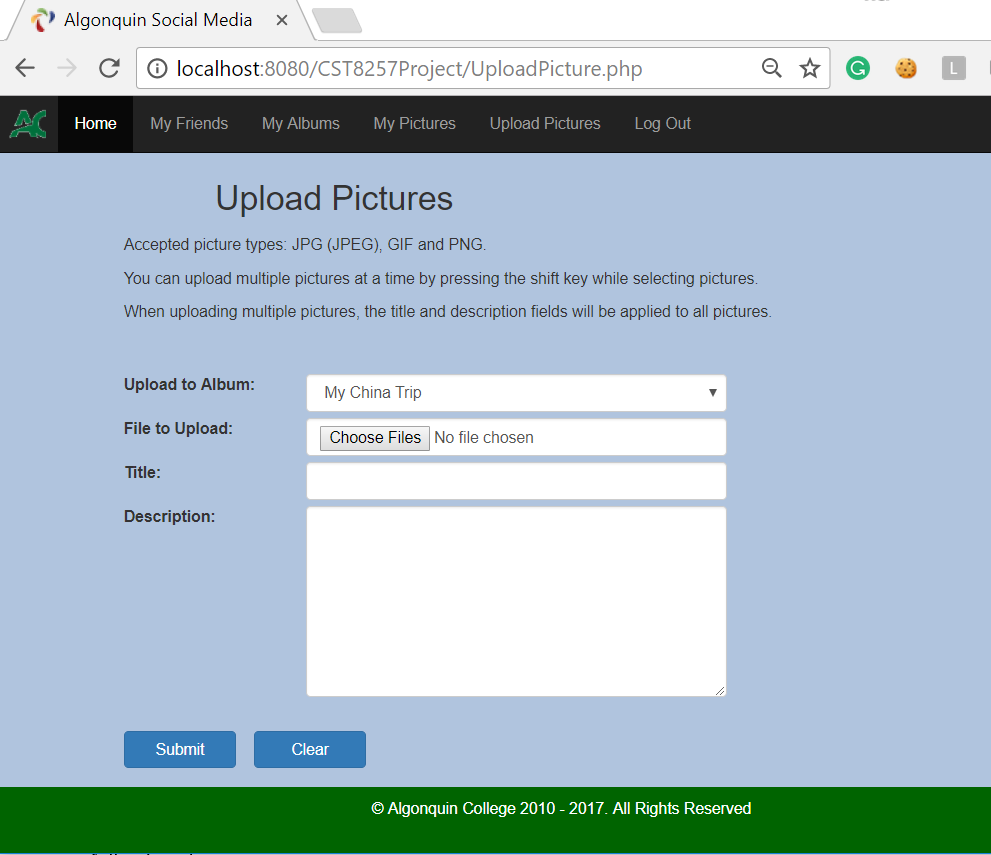


The page also has following links:

* Create a New Album – link to **AddAlbum.php** page
* Album Titles – All album titles in the album list are links to **MyPictures.php** page once clicked, the user is brought to **MyPictures.php** page with the clicked album as the selected album in the dropdown list, see **MyPictures.php** section for details of MyPictures.php page.
* Delete – a link button to delete the album. Once clicked, the system will prompt the user to confirm that all pictures in the album will be delete with the album.

1. **UploadPictures.php**

The user uses this page to upload pictures into one of his/her albums.

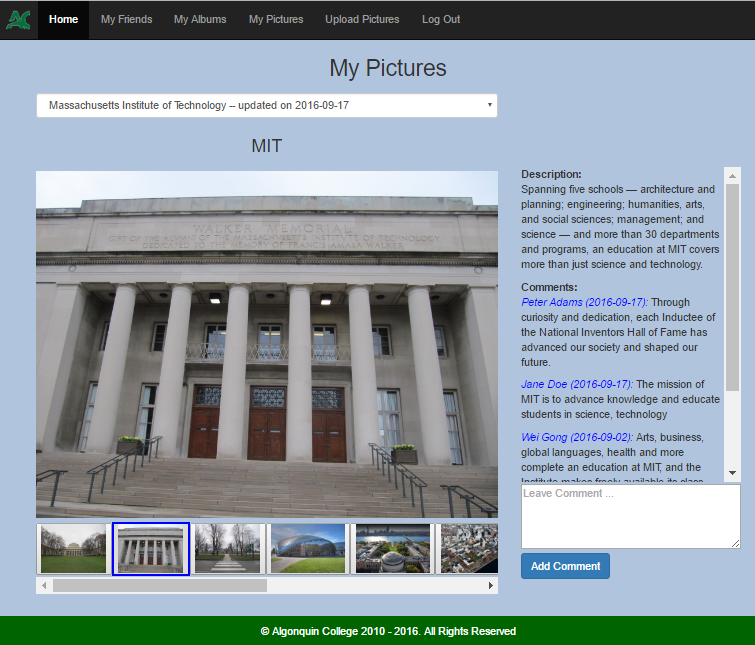


The user can upload single or multiple pictures at a time. Optionally, the user can specify title and description for the pictures to be uploaded. When uploading multiple pictures at a time, the specified title and description apply to all the pictures uploaded.

1. **MyPictures.php**

This page is for the user to browse and manage his/her pictures. The page contains the following elements:

* A dropdown list for the user to select an album to browse its pictures.
* A picture area showing the picture selected when the user clicks its thumbnail.
* A thumbnail bar displays all thumbnails of the pictures contained in the album. When a thumbnail is clicked the picture area displays the picture the thumbnail represents. The thumbnail of the picture currently in display should be highlighted with a blue border.
* A description and comment area showing the picture’s description (if exists) and comments (if any). The comments are ordered chronically from the newest to the oldest.
* A text area for the user to leave a comment. The user can write a comment and click the **Add Comment** button to leave a comment about the picture.



Picture description

(if exists)

Picture comments

(if exist)

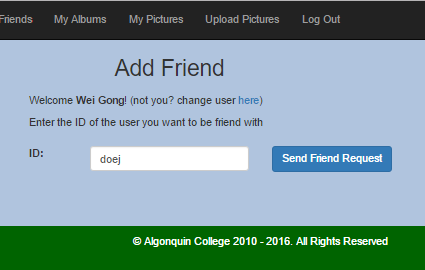
When the user hovers the mouse over the picture, four icons show the actions the user can perform to the picture:



* Rotate the picture count-clockwise
* Rotate the picture clockwise
* Download the original picture
* Delete the picture from the album

1. **AddFriend.php**

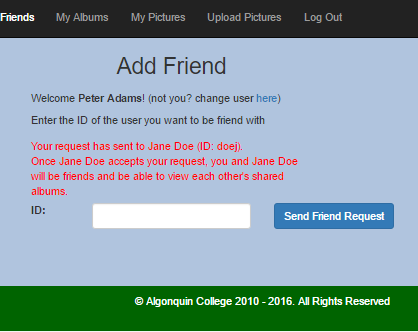
The user can use this page to send a friend request to another user by enter the other user’s user ID and click the **Send Friend Request** button.



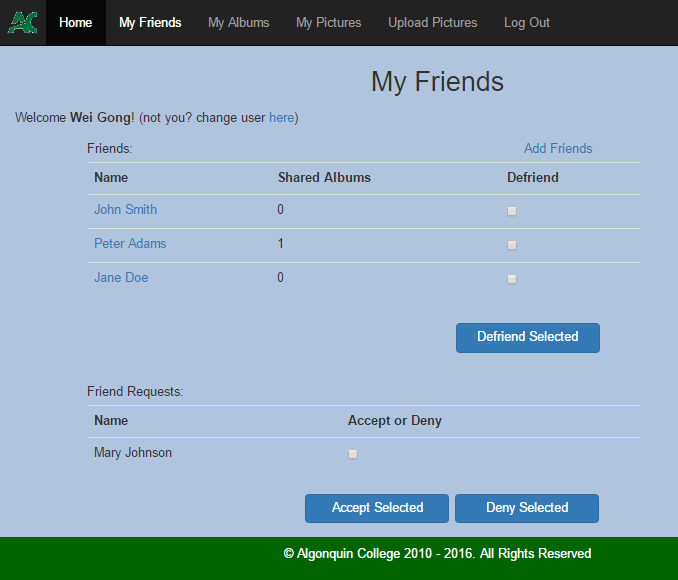
A friend request has to follow the following rule:

* The entered user ID must exists.
* One cannot send a friend request to himself/herself.
* One cannot send a friend request to someone who is already his/her friend.
* If A sends a friend request to B, while A has a friend request from B waiting for A to accept, A and B become friends.

If the request passes the above rules, the user will get a confirmation message to confirm that the friend request has sent to the specified user:



1. **MyFriends.php**



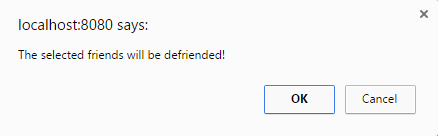
This page lists the user’s friends and friend requests. For each friend, it shows the number of shared album of that friend. The user can perform the following action for each friend.

* Click the friend’s name to view the friend’s pictures in the shared albums
* Select the checkbox and click **Defriend Selected** to remove him/her from the friend list.

For each friend request, the user can check the checkbox and click **Accept Selected** to accept the friend request to become a friend of the requester. Or click **Deny Selected** to decline the friend request. Once accepted or declined, the friend request will be removed from the friend request list.

The page also contains a link Add Friends to page **AddFriend.php**

When Defriend a friend or Deny a friend request, the user should be given a proper warning:



1. **FriendPictures.php**

This page is similar to MyPicture.php excepts:

* The album dropdown list only has the shared albums in the list for selection.
* No action icons will show when the user hovers the mouse over the picture.

