**Tai Chi Martial Arts**

**Milestone 1 Part A**

Team member names below

Arjay Nguyen,

Boya Zhou,

Kevin Tran,

Joseph Hu

Dec. 1, 2014

CS 174

Group 2: Team ABJK

**Assigned domain: fun\_video2.sql**

Contents

Individual Contribution Summary. 2

Introduction. 3

What have been accomplished in this milestone. 3

Data correction and preparation. 3

Restoring Data. 3

Data processing. 4

Data correction. 4

Tagging videos. 4

Video info modifying site. 4,5,6

Login/Registration. 7, 8

Cookies for storing login info. 8,9

Favorite videos. 9 - 12

Additional info. 12, 3

# **Individual Contribution Summary**

|  |  |  |
| --- | --- | --- |
| Student Name | Technical part | Report drafting |
| Arjay Nguyen | ·implemented ability to edit database information w/o PHPMyAdmin access  .implemented hide edit feature unless user is logged in  .pull in database information and add existing info to the edit page to allow user to see existing values  .minor CSS changes | • added report template  • added Individual contribution  • contribute to Introduction  • wrote section Video Info Modifying site |
| Boya Zhou | Implement part B & C.  Create log in page allow user to log in and register. Provide a checkbox for users to opt in or opt out for system to store user login info with PhP Cookies.  Design the the login page using css.  login.php, process.php  login.sql | Added Individual contribution  Added explanation and images on how to implement log in and register, also use cookie to store log in info. |
| Kevin Tran | * Implemented part D, ability to add and remove favorite videos. * Implemented sessions for users; pull and store user data * Modified login table, add favorite videos. * Displayed favorites on homepage * Edited view2.php, index.php, login.php, process.php, addFavorite.php | * Added Individual contribution * Added to favorite videos * Added images to show process |
| Joseph Hu | .Performed Part A with data manipulation and corrections.  .Watched every video to make sure the data was correctly reflected.  .Assisted in creation of favorites module for logged in users. | .Data correction preparation  .Favorite Videos |

# **Introduction**

## What have been accomplished in this milestone

Our work is hosted on: [http://www.sjsu-cs.org/classes/cs174/sec1/tran/projectmilestone1/index.php](http://www.1stsite.com/)

Whose web and database account are used in hosting this website: We are working off of Kevin Tran’s Bonus HW submission, using the fun\_video2.sql file as assigned to us in the Group Homework description.

In this milestone we have implemented features A-D according to the project description. To summarize, we have added the ability for a user to log in. Once they have logged in they are able to access new features that are only available to logged in users(i.e. saving favorites, edit database). We have also cleaned the database items and removed any discrepancies in videos that do not meet the requirements for our database.

Accounts for testing login info:

1. [arjay@email.com](mailto:arjay@email.com) - password1

Describe what you have accomplished in bullet form:

* Cleaned database items
* Implemented login feature
* Implemented edit feature
* Implemented favorites feature

# **Data correction and preparation**

## Restoring Data

The recovery of the data file was simply an execution of the given .sql file and executed into the phpmyAdmin database manager. The data however recovered was full of errors and inconsistencies. In this case, it was the fun\_video2.sql.

## Data processing

The removal of data items such as duplicates and view count was an easy task as we used SQL statements to filter and weed out any data that was not consistent with what was asked of us such as a delete statement for viewcounts < 50000.

## Data correction

The discrepancies and data correct was a more laborious task as each video was either very or a little different from one another. Thus, it could not be assumed that we could treat any video the same. Therefore, each video was watched and edited manually to make sure the data was as consistent as possible.

## Tagging videos

The tagging of videos could be done consistently with certain amount of assumptions. Certain videos were created by the same user or contained the same instructor so tags were moved over from one to another. Certain specialized tags though needed customized for certain videos so each video also needed to carefully combed through.

## Video info modifying site

Instead of adding a script to the homepage, we thought it would be more practical to add an edit button to the side of each table entry, provided the user is logged in. We also pull in the current values for that database item to display to the user on the edit page. Once that’s done we created a SQL UPDATE query using the new values and query the database.

1). First things first, we need to add the edit table header and the edit button to the video table, but we need to make sure it’s hidden if the user is not logged in.



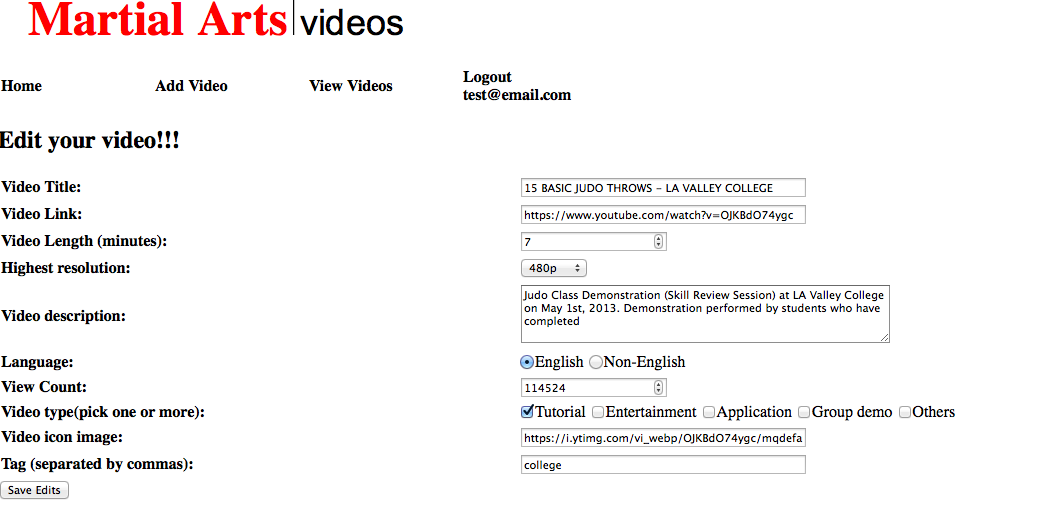


We accomplish this by calling $\_SESSION[‘username’] to check if the username variable is set in the session. Then we echo the appropriate html code to display the Edit header/buttons. We will also pass the id of the database item when the button is clicked on.

2). Then we create edit.php, which the Edit button will call once it is clicked on. In this php file we need to create an html form, similar to the addVideo page. The only difference is we need to pull in the database info using the id passed in $\_POST. We use the id to query the database for the rest of the information, then we echo the html code and add in the current values as defaults to the form items.



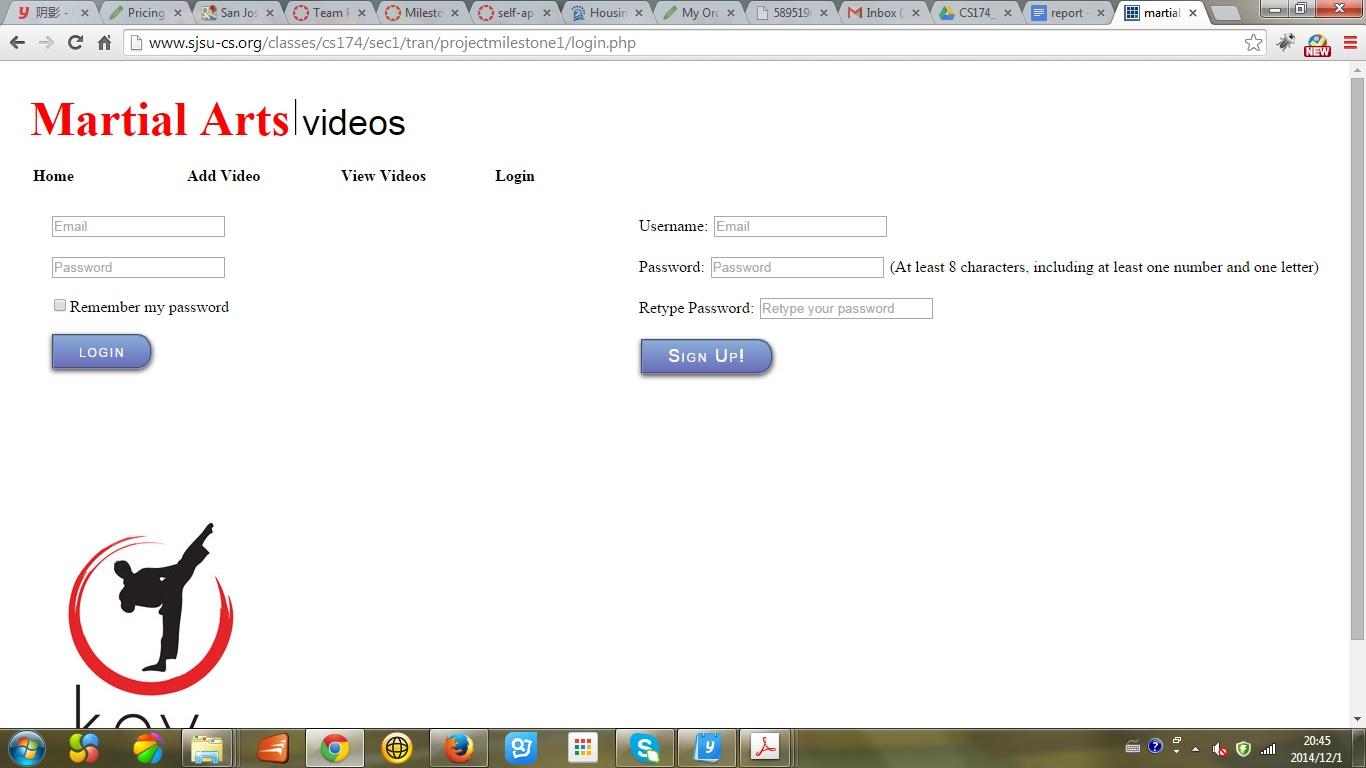
The result is a page that looks like this



3). The last task is to grab all the values in another php script, and send a query to update the database. This task was very similar to creating an entry in the database, but we use the UPDATE query instead.



# **Login/Registration**

We create login.php for both log in and register an account. In our database, we use login.sql to create a table to store all usernames and passwords. In login.php, there are 2 forms. One for user log in, one for register. 

Once user fill in all the information and click on sign up, It is going to handle the form by login.php itself. If the password meet the requirement, it is going to insert the username and password to the database, and jumps to the user’s homepage. Otherwise, it is going to stay in login.php and ask user to sign up again.

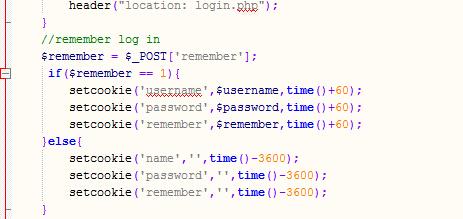




For the user sign in form, once user fill in the form and click on sign in. Process.php is going to handle the form. It checks if there is a match in the database. If it does, it redirect to user’s homepage. Otherwise, it jumps back to the form.

# **Cookies for storing login info**

We use cookies to store users’ login info. If use check the checkbox in the log in form. It is going to create a cookie to store the username and password. And automatically put in in the login form.



# **Favorite videos**

Sessions were an important part of this module. Once the user logged in, the session needed to continue throughout his/her experience to be able to record the data. When the user successfully logs in, a session is started. The script does a query to pull the user’s list of favorite videos from the database and stores it into a session; it also stores the user’s username in another session variable. The stored user name is what we used to check if someone is logged in. Once logged in, if there are any favorite videos, they are displayed on the home page. In the favorite videos module, the user had to have a checkbox to make sure he/she could check the videos enjoyed. The session needed to record the answers and pass it through different pages. The array also needs to be translated into mySQL to be stored in the database which presented a challenge as mySQL did not have an array datatype.



Start session - username and favory array is loaded into the session



Check boxes - if liked it’ll display as checked

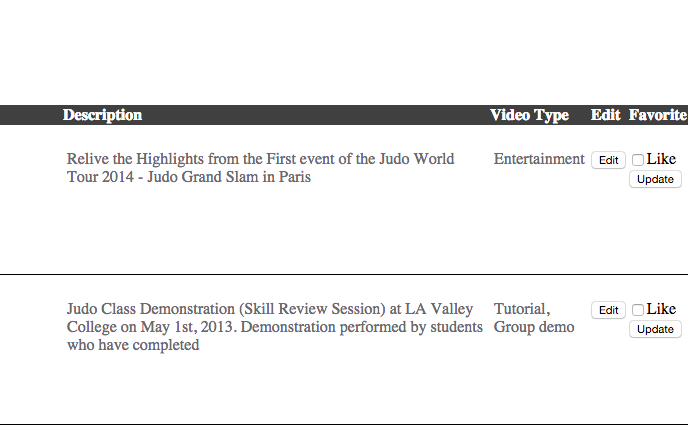


Favorite Videos on Homepage - up to four favorite videos are displayed on the home page



This code adds or removes the video id to a session array and updates the database. Notice how we had to serialize the video id array.

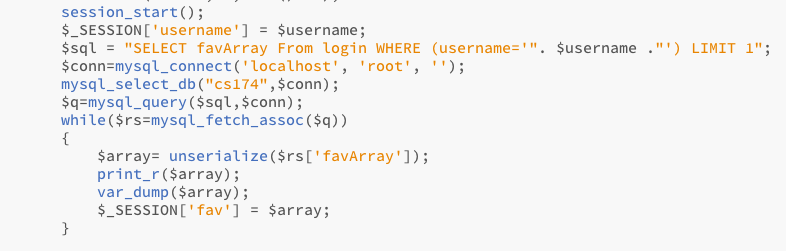
The other functionality that gave us unlikely trouble was how we were going to remove or add favorite videos quickly through the user interface. We did that using the <checkbox checked> tag and PhP isset boolean to help resolve this problem.



Favorite checkboxes- user checks like and update.

# **Additional info**

For the favorite video arrays, we had to serialize the arrays to store them as a String in the mySQL database. When we retrieved it, we had to unserialized the data and store it in a session variable called $\_SESSION[‘fav’].



Retrieving array



Storing array