Team Project – Prototype Report

November 30th, 2014

Tanya Brobrowski
Sara Cavasotto
Nicole Mackin
Evan Sherlock-Hubbard
Joshua Allin
Anjali Dutt
Fred Pilon

Presented to Chris Joslin at Carleton University

1. Tanya Brobrowski

Objectives

My main objective was to create an events page, or at least a basic skeleton of one, and also try and simplify some of the code of the other pages. I decided to make a basic events page with some of the layout and also make a navigation bar document so that it could be included in the other pages and easily editable later.

Methodology

Events Page

For the Events page I planned to have on the left side a list of the user's events and on the right side their calendar. Their calendar would be able to be switched to other months and days with events in them would be coloured.

Pseudocode

```
    Left Column
·   Take list of events for user and generate them in a list (with icons)
    Right Column
·   Generate Calendar table based on current date
·   If an event in user's event list is on the same day as one of the days in the calendar, highlight the date (change background color of <td> for that entry)
·   If a day on the calendar is clicked, call a function that creates info about that day below the calendar (list any events for that day) and also give the option for the user to create a new event for that day (button)
```

Navigation Bar

The navigation bar was simply going to include the html needed to create the navigation bar at the top of each page. It did not really involve any ingenuity on my part. All that needed to be done was put the code into another document and then include it within all of the other pages. The purpose of this was so that if we needed to change the buttons later it could be easily done and we would not have to change the code in every single document to do so.

Conclusion

In the end, I was able to create the main skeleton of the event's page and create the navigation bar for easier development later. The layout isn't quite right yet for the events, but it will soon be fixed. In the future, I plan to finish the event page along with its data and functionality and also create more art assets for the website so that it looks more visually appealing, as well as help with anything else that I might be needed with.

2. Sara Cavasotto

Objectives

My objectives were to organize everyone's contributions into a basic HTML/CSS layout preserving the likeness from page to page. I have created the navigation and the profile layout and integrated josh's messaging into that layout. Working on the layout is also making it easier to work on some of the PHP because they now have somewhere to apply it too. And I have begun to pseudo code multiple pages we still need. and Ive coded the edit page to pull the needed information into editable text boxes.

Methodology

Listed are the pages I will continue to work on implementing:

edit.php

To give form that contains text boxes for About Me, Dream, Music, and Profile Pic. When this page is opened user id is used to put answers from user table into the textboxes to allow manipulation instead of having to rewrite everything. (This part is done)
When user is done it overwrites that stuff in the table with the new information. It it starts displaying the new information on the website as soon as you click a button that implements it.

Friends.php

This page will have a list of friends with their picture and a limited amount of characters from their about me. This layout will be repeated till the page is full of they run out of friends to display.

Action Plan

First step would be to in the many to many table (friends) using the players ID pull out the UserID attached to its id on both sides (User1 and User2)
In the User table using the ID numbers pulled out get their profilepic information and about me information and print it in and appealing way. Keeping the Id attached to each print using a php variable and making the name clickable to jump to their page from there'

Conclusion

I have focused on the user interface design, by cleaning up redundant code and unifying the overall look of the webpage. This included creation of the art, organizing the layout, and making sure all code follows a similar pattern allowing for changes to be made, and keeping all code consistent.

I have coded a system that reads your user ID (right now I just have a variable but that is not hard to change to reading a cookie with the current user's id), compares it to the id of the profile, and allows for editing certain values based on it. I have finished the coding for pulling the information and I still need to code replacing the information with the new values.

3. Nicole Mackin

Objectives

One of the requirements for the group project is to have a cooperative game that allows two users may play together to complete a common objective. I was placed in charge of the creation and basics of the game, as displayed in “game.php”. I was also tasked in creating the basic SQL query to determine whether users are friends, used in the majority of the pages found in our website, and the database layout.

Methodology

For the cooperative game, I planned to make a game that allows users to attempt to find a common word with their friends. The goal of the game is to guess a word that is related to the last two words given, and guess the same word as your friends.

This is accomplished by writing a code that accesses your friends names, and displays any games related your your friendship. This allows for players to only play with user that are their friends. The user may then follow this link to their game within the game.php page. This then allows the player to look at the previous rounds of words. If there are no winning matches and the user has not yet played a word, a textbox is displayed allowing them to play their word, thus saving it into the GameRound table, under the next available round. This allows for a new round to be created if no other round of its ID exists, and to add the word into the existing game round if it is found. Once the round is played, the user no longer has to option to submit more words until the other user plays a word.

Once both words are received and a user opens up the game, the latest rounds words are checked for a match. If the string compare returns true, the text box is not displayed and displays a “game won” text. If not, the process of inputting data is restarted and the user continues in an attempt to win.

In order to improve upon this concept, a way to hide the round from play until both words have been read into the database must be put into place, as to keep users from being able to read their friends words before inputting their own. It would also be beneficial to code a way to edit words before the other user has played theirs, in case of possible mistakes.

I used the friendship code used in the game in the rest of the pages as a base, allowing us to ensure consistency throughout the website’s SQL. I designed the basics of the database, determining the type, name and length of the elements, as well as their relationships set by their primary and foreign key values.

Conclusion

The main body of the cooperative game is complete, and the code used in determining relationships between users is kept consistent throughout the project. The tables needed for all of the pages have been determined, and their relationships have been set.

4. Evan Sherlock-Hubbard

Objectives

My Objective has been to improve the functionality between the front-end website, and the information on the server (things like user profile information and the editability of the events page) As well as help with basic add-ons to the forms such as double-checking that information submitted to a server is in the correct format (Ensuring a user's submission for their birthday is a date, that their email is an email address, etc.).

Methodology

Submission Error-Trapping

Making sure that fields are required to be filled before the form can be submitted as well as making sure that they are in the right format isn't too difficult with HTML's newer form modifiers, simply marking them as required or setting their type to email or date.

My other goals thus far have not had much success and currently any of the non-simplistic functions I've created cause the web page to lose functionality drastically, leading to their removal until they are error-free. This will change in the final iteration of the website. The login page is ready to have the information submitted, but the server does not yet do anything upon submitting the form.

Conclusion

Right now my part in this project has not been significant, but that will change as the website becomes more functional. The final version of our website will have the functionality to let the user send information like events and profile changes to the server and have it handle the necessary formatting, but thus far the website is not at that point yet.

5. Joshua Allin

Objectives

One of the objectives set out for our website was to have a functionality set in place that would allow for users to leave text posts on other users' profile pages and for those posts to also be available to comment on. I was allocated the task of completing this objective.

Methodology

In order to accomplish the task of having a profile commenting system, the task was split into two parts: text posts on the profile page, and their comments.

To set up the page to allow text posts, a form with a method of POST was set up, containing a textarea and a submit button. Using php, the page then read the characters submitted from the textarea, and the time submitted and uploaded them to the database in 'phpMyAdmin' using an INSERT query. The page then displayed all of the text posts from the database onto the page. If no text posts are present in the database to be displayed, then an error message was displayed informing the user that there were no text posts to be displayed. This was done using an if statement and mysql_num_rows.

Setting up the page for comments to be left on the text posts was handled in much the same way, except that they only appear on the text post they were intended to be left on. This was handled by using php. An input of type 'hidden' was used, passing the value of the text post along with the rest of the comment's information. Then, by including a where clause in an sql query that only drew them where their text post was present, only the appropriate comments were shown.

Conclusion

The only aspect of the text posts feature is the use of AJAX to dynamically update the page without the scroll being sent to the top of the page, and to set it up to work with specific profiles depending on which account with which the user is logged in.

6. Anjali Dutt

Objectives

One of the objectives of the website is to be able to upload an image from the user's files to use as a profile picture of an account.

Methodology

The first step was to create separate columns in the users table for the profile picture, since it should be kept separate from other images that could possibly be uploaded to the website. This allows us to create and insert the image into the database at the same time a new user is created. This will also allow us to retrieve the image from the database to use on the profile page. Code from tutorial 9 was used to upload images to the database.

Conclusion

What remains is to create the rest of the login page so that each profile pic has user information to go with it. The upload button also needs to be combined with the create user button so that both are done at the same time instead of separately.

7. Fred Pilon

Objectives

One of the functions for our website is to have the ability to send a private message to a member of your friends list and have a conversation. The page would allow you to choose what friend you wish to speak to and write a short 255 character message and send it to them.

Methodology

In order to send message to other profiles, a relationship needed to be formed between users (a recursive relationship) and a table that would save the messages and remember who was the sender and who was the receiver.

The first thing to do was to create a basic relationship that assumed that the user was of the ID number 1 at all times. This will be changed for the final version so that we can read which user is currently logged in and dynamically find out who is on their friends list, but for now, it will be kept this way for testing purposes.

After creating a relationship between users and the Messages table, a simple POST method was implemented that allows us to upload messages to the database using an INSERT query. After confirming that our messages were uploading properly, we allowed our drop down menu to not only dynamically display the names of friends through the use of a while loop, but to also dynamically change the receiver ID that will be receiving the message.

Conclusion

What remains is to be able to pull past messages from the database in order to view the entire conversation and to display which user sent the message, and to better center to messages on the webpage.