Homework Assignment 5

Objectives:

- To gain more experience in serverside programming.
- To get familiar with basic PHP.
- Learn about the unix timestamp.

Calendar Part I

Introduction

In all web 2.0 applications, creating web pages dynamically is essential. It is very common that a web 2.0 website retrieves data from a database and then renders a new webpage based on that data. In this assignment, you will create a blank calendar using PHP that will be used later in HW6 to display upcoming events that are retrieved from a database.

Description

Your PHP code should render a calendar page similar to the one shown here. The top of the calendar states the day and time being displayed. The calendar is a table where the first column lists hours starting from the current hour to a final hour (where the final hour is the current hour + some number between 0 and 12). For example, if the time is now is 5.32pm, the first hour should read 5.00pm and the last hour is determined by a variable called hours_to_show. If hours_to_show is set at 12, then the last hour shown in the column should be 5.00am. The subsequent columns correspond to people. In the example, I have used 3 people: You may have as few as two or as many people as you like.

Directions

- Start by viewing the source of the XHTML file of the sample calendar to see the XHTML output of the PHP file. This will give you an idea of what the output of your PHP program should be.
- Be aware that the requirement is that the number of hours being displayed is set by a variable. In practice this variable will be set to 12 at the start, however, this means that the number of rows in the table cannot be assumed to be known and you cannot simply write XHTML for 12 rows.
- I have prepared a CSS file for you that you can use to style your calendar. You can get the CSS here. This is just for your convenience. You may choose to write your own CSS if you so desire. When using my CSS file please note that certain elements need to be given appropriate ids or classes. View the source on the XHTML file of the sample calendar to see the XHTML output of the PHP file and to see what elements need to be given class or id labels.

General notes

- Make sure that your file has correct permissions set. php files should have permissions set to 755 or if you prefer: -rwxr-xr-x. Which means that it should be: **read** for owner group and other and **write** for owner and **execute** for owner group and other.
- For debugging purposes you can change the top line to be: #!/usr/local/bin/php -d

display_errors=STDOUT and you will get error output on your document, which makes finding syntax errors much easier.

Requirements

- Your program should be called calendar.php and should reside in your public directory along with a CSS file called callendar.css that styles it.
- When your program is finished, create a text file called calendar.txt and copy and paste the contents of your calendar.php file into calendar.txt. Put calendar.txt in your public directory as well.
- Your calendar must be generated by PHP script.
- By default when your page is loaded it should display current day, date and time as shown in the example.
- Your program must use a function called get_hour_string to get the hours for the first column of the calendar table. This function takes a single argument of a timestamp and returns a string indicating an hour. For example, the value returned might be: "8.00am".
- The table is rendered by PHP the number of rows displayed is determined by a variable hours_to_show, which you may set equal to 12.
- The rows should aleternate in background color.

Validate the output of your PHP program. Go to w3c's validator page and give the url of calendar.php and w3c will validate the resulting XHTML.

Make sure the following files are uploaded to your `public_html` directory and CCLE:

- `calendar.php`
- `calendar.txt`
- `calendar.css`

Grade Breakdown:

Criteria	Points
Generates the date and time at the top of the calendar correctly.	20 points
Alternating colors for the rows are correctly rendered and aesthetically pleasing CSS.	20 points
The function get_hour_string functions correctly and the hours are displayd correctly.	30 points
The rest of the table is correctly rendered by PHP and continues to be correctly rendered if hours_to_show is changed to something else than 12.	30 points
Total	100 points