

# Homework Assignment 6

Objectives:

- To gain more experience in serverside programming.
- To learn about processing forms with PHP.
- To learn about using PHP to control a SQLite3 database.

## Calendar Part II

### Introduction

In HW #5 you used PHP to create a blank calendar. In this assignment the goal is to first write events into a SQLite3 database and then read these events from a SQLite3 database and display on the calendar the upcoming events.

### Description

Your PHP code in the file `update_database.php`, should process the data a user fills in on the `create_event.html` page. Then your program `calendar2.php` should display the events in the correct time slots and for the correct person. The functionality of your programs should be similar to the example on the homework page. For example, consider that we set in the `create_event.html` the person to be Joe and the date to 5-20-2011 and time is set as 15:00. The user writes the event title as: "Soccer practice" and event message: "Bring the ball". Then the `calendar2.php` should show for 5-20-2011 in the Joe column in the 3:00pm slot: "Soccer practice: Bring the ball". You will also "upgrade" your `calendar.php` from last week to display at the bottom of the calendar three buttons labeled: "previous", "now" and "next". The button labeled "previous" should move the start hour displayed back by 12 hours. The button labeled "next" should move the start hour forward by 12 hours. Finally the button labeled now should display the calendar with the current date and time.

### Directions

- You have almost two weeks to do this assignment, but it is hopeless to start on this assignment a couple of days before its due date.
- Download the necessary files and place them all in your public folder. Set the correct permissions for all files.
- Note that you should modify `create_event.php` slightly so that the names of the people match your calendar from the previous assignment.
- In few days I will provide for you a solution to HW5 that you can use as your starting point if you so choose. Note that I have also provided you two css files that style the `create_event.html` and `calendar2.php`. If you prefer, you may write your own CSS.
- You will write a php file called `update_database.php`. This program will receive the form data when the user presses submit on `create_event.php`. It will then process this data and save the data into a SQLite3 database.
- Your program `calendar2.php` is much like your calendar program from HW5, except now it will also read data from your SQLite3 database. For each slot (person and time) your code should call a function called `get_events($person, $timestamp)`. This function will return a string containing all events that should be displayed in the time slot in question.

### General notes

- Make sure that your file `calendar2.php` has correct permissions set. php files should have

permissions set to 755 or if you prefer: -rwxr-xr-x. This means that the settings should be: **read** for owner group and other and **write** for owner and **execute** for owner group and other.

- If your top line reads as: `#!/usr/local/bin/php -d display_errors=STDOUT` you will get error output on your document, which makes finding syntax errors much easier.

## Directions for your database

- Your database is called `dbyourusername`.
- Your table should be called `event_table`
- Your table should contain field `time` which is a timestamp it is going to be an integer with perhaps as many as 12 digits.
- Your table should contain a field `person` which has character data. (Reserve some reasonable amount of characters for a name.)
- Your table should contain fields `event_title` and `event_message`. Both contain character data max of perhaps 300 characters.

## Requirements

- Your program should be in two files called `calendar2.php` and `update_database.php` those files should reside in your public directory along with CSS files called `calendar.css` and `create_event.css`.
- When your program is finished, create a text file called `calendar2.txt` and copy and paste the contents of your `calendar2.php` file into `calendar.txt`. Similarly create a text file called `update_database.txt`. Put `calendar.txt` and `update_database.txt` in your public directory as well.
- 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_event` to get the events for the calendar to display.
- Your program must function as in the example given.
- You must have the three buttons: previous, today, and next that function in the way described above.

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` and `Submit` directories:

- `calendar2.php`
- `calendar2.txt`
- `update_database.php`
- `update_database.txt`
- `create_event.html`

## Additional instructions

- Write your code for `update_database.php` so that the date in `create_event.html` can be entered in form MM-DD-YYYY. eg. 11-02-2011.
- Note that your event text can not contain apostrophies quotes or single quotes. (This is a technical point. Ask me if you are interested in the why.)

## Try out the sample

Try out this sample code that demonstrates how the program is supposed to run. Keep it clean or the program will be removed from online and only be available by request during my office hours.

- [create\\_event.html](#)

- [calendar2.php](#)

Grade Breakdown:

Criteria	Points
update_database.php correctly writes events into the database.	30 points
get_event_string functions correctly.	30 points
The events are displayed for correct person at correct time.	20 points
Everything validates and functions correctly and looks presentable	5 points
The three buttons "previous", "next" and "now" function as described.	15 points
Total	100 points