IT 207
Applied IT Programming
Lab Assignment 2 | Control Structures, Web Forms

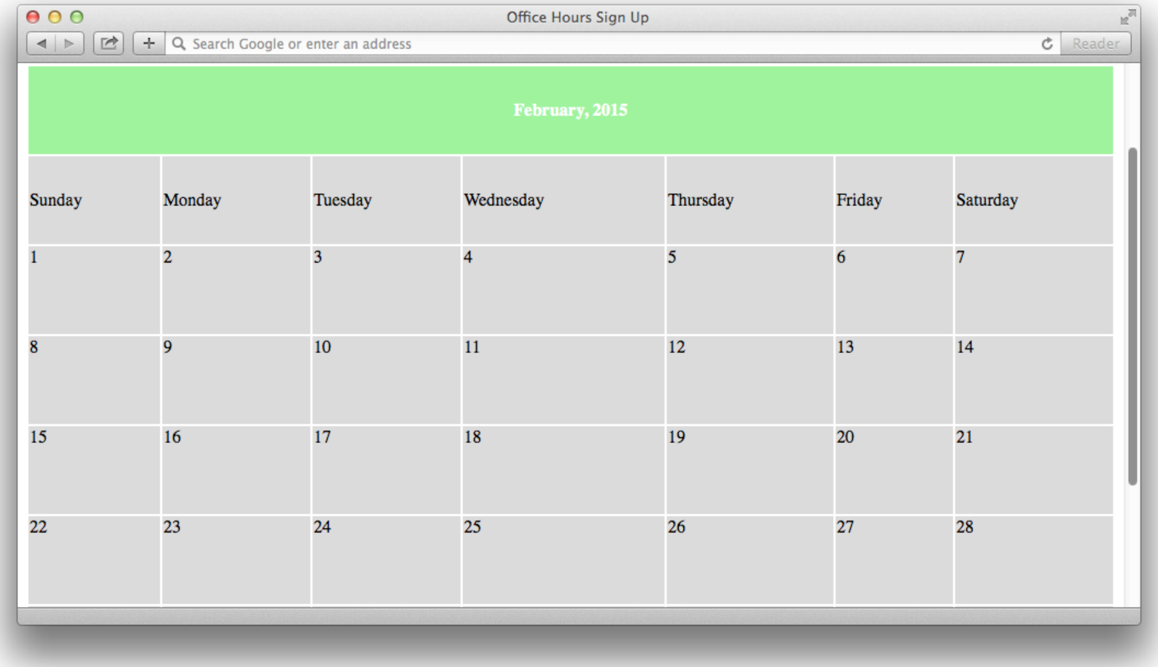
Office Hour Booking Form

In this assignment you will build an online appointment book for the current month. First you need to generate a static web-based calendar for this specific month. Then you will add the ability for an individual to specify days of the week and times within those days where the professor is available for bookable meetings. Based upon the days and times chosen you should provide a calendar that allows a student to select a time and book a meeting with the professor. The use of the calendar will allow the student to more clearly see what dates and times are available and which days of the week the dates correspond to. Calendars such as these are found in business applications like Google Calendar, iCloud Calendar, and Microsoft Outlook.

Design Overview and Requirements

As part of the first submission point, you should generate a static calendar for the current month. The current month used for the header and display can be determined in your code by using the `date()` function. To be eligible for grading you should construct the calendar using one or more control structures nested inside a calendar .php script file.

An example of how the calendar may appear (if February 2015 were the current month) is as follows:

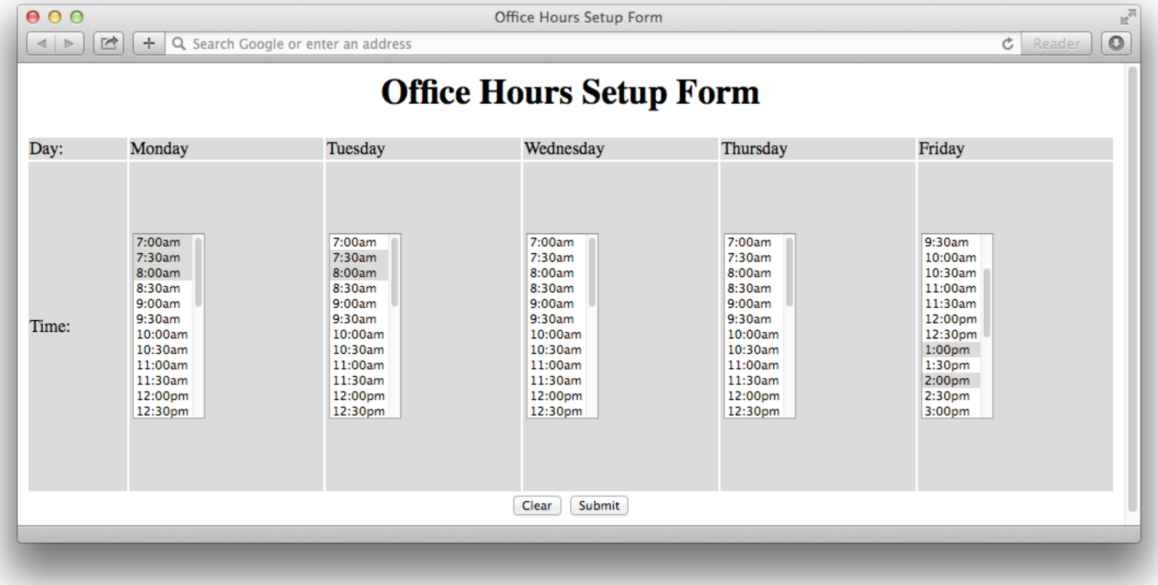


February, 2015

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

For this assignment, in order to make the calendar usable for office meeting scheduling you should provide an initial setup page that allows an individual to select days of the week between Monday and Friday where he/she is available for meetings. Within each of these days you should provide 30-minute blocks that can be chosen as times when meetings can be scheduled. You should allow one or more time blocks to be selected whether in succession, or sporadically throughout that day of the week. Available times for choosing should be between 7:00am and 10:00pm. The days and times chosen will apply to the whole month that is eventually displayed.

On the next page is an example of the setup page output you should have before formatting:

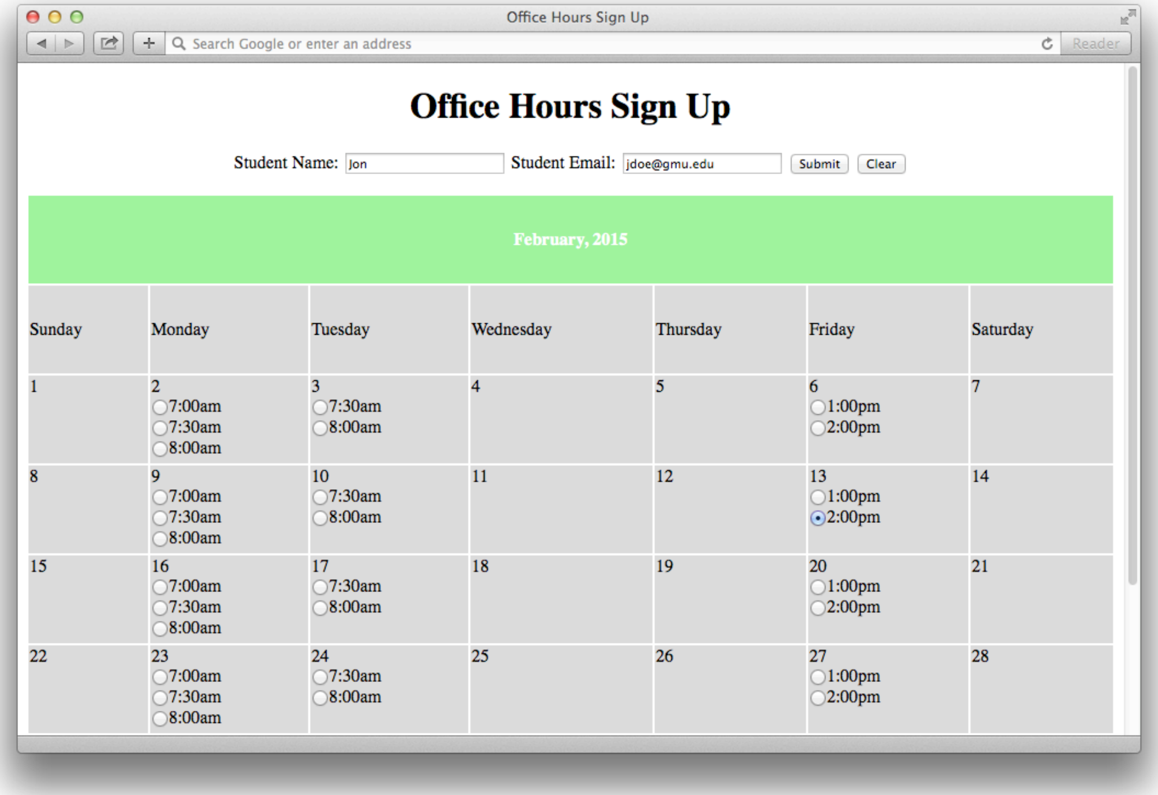


Office Hours Setup Form

Day:	Monday	Tuesday	Wednesday	Thursday	Friday
Time:	7:00am 7:30am 8:00am 8:30am 9:00am 9:30am 10:00am 10:30am 11:00am 11:30am 12:00pm 12:30pm	7:00am 7:30am 8:00am 8:30am 9:00am 9:30am 10:00am 10:30am 11:00am 11:30am 12:00pm 12:30pm	7:00am 7:30am 8:00am 8:30am 9:00am 9:30am 10:00am 10:30am 11:00am 11:30am 12:00pm 12:30pm	7:00am 7:30am 8:00am 8:30am 9:00am 9:30am 10:00am 10:30am 11:00am 11:30am 12:00pm 12:30pm	9:30am 10:00am 10:30am 11:00am 11:30am 12:00pm 12:30pm 1:00pm 1:30pm 2:00pm 2:30pm 3:00pm

The chosen times on the Office Hours Setup Form should be passed to the calendar .php file using the POST method. In the .php file retrieve the passed values from the superglobal. These passed values may be primitive variables and/or arrays depending on how you set up the Office Hours Setup form. You should display a current month calendar showing each day's selected meetings times as bookable times.

Below is an example of the output you should have before formatting:



Office Hours Sign Up

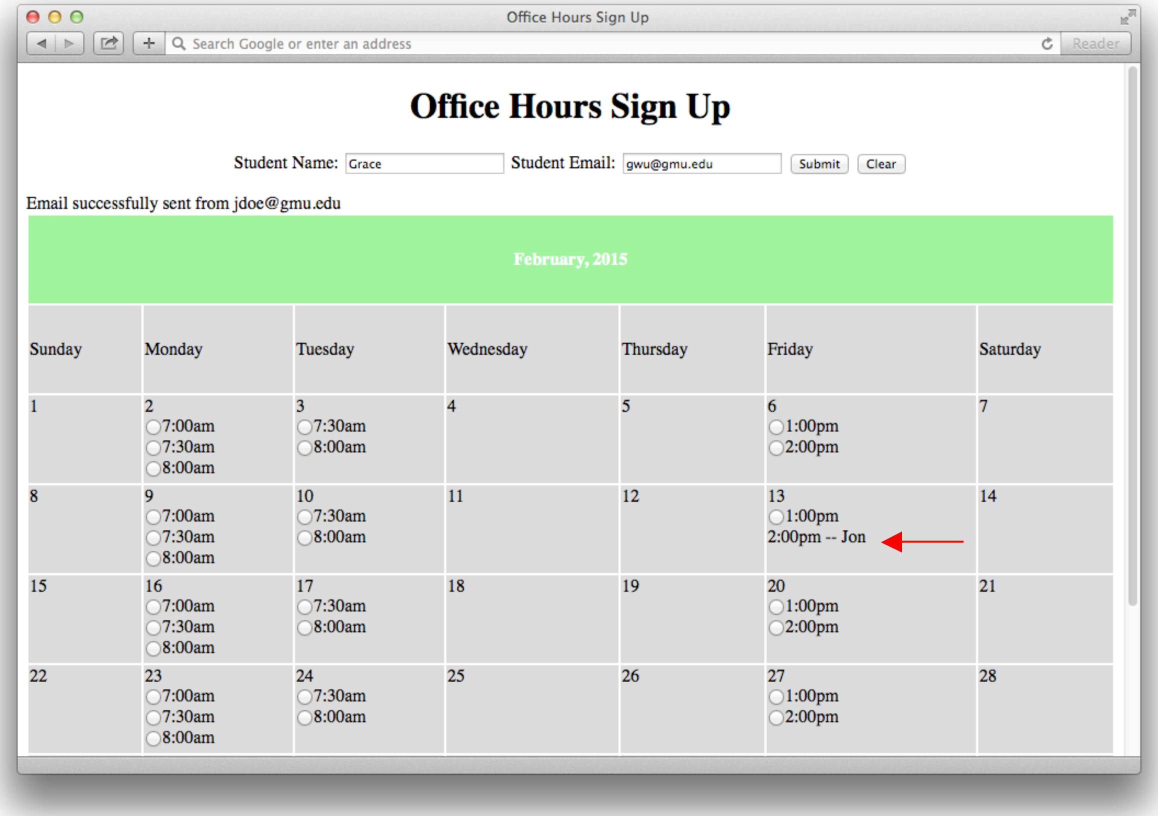
Student Name: Jon Student Email: jdoe@gmu.edu

February, 2015

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2 ○7:00am ○7:30am ○8:00am	3 ○7:30am ○8:00am	4	5	6 ○1:00pm ○2:00pm	7
8	9 ○7:00am ○7:30am ○8:00am	10 ○7:30am ○8:00am	11	12	13 ○1:00pm ●2:00pm	14
15	16 ○7:00am ○7:30am ○8:00am	17 ○7:30am ○8:00am	18	19	20 ○1:00pm ○2:00pm	21
22	23 ○7:00am ○7:30am ○8:00am	24 ○7:30am ○8:00am	25	26	27 ○1:00pm ○2:00pm	28

With the meeting times visible, you should provide a way for a client to provide their name, email address, and select a time to meet with the individual. This functionality should then show them signed up for the selected meeting time while also displaying the remaining available meeting times. Hint: You can send a form to itself (the same file that has the form) and pass some of the values that the form may now need via hidden form fields. Hidden form fields can have a name and value assigned to them. As part of signing up for the meeting time, your .php file should automatically send an email to the professor from the student with the meeting time information. The `mail()` function should be used and if the function returns True, then you should assume that the email was successfully sent to the professor. You do not need to check with the professor whether they received the email, just check the return value from the `mail()` function.

An example of what a booked meeting time by a client looks like is provided in the second week of the calendar of the example diagram below. The action of booking a meeting is done by having the client type in his/her name, select an available meeting time, and then clicking 'Submit'. Once submitted, the selected time should be replaced on the calendar with their name for that chosen time along with an indication of whether the email was successfully sent. While subsequent bookings of meetings by students do not require you to retain the previous students client's booked time (as this is not of concern to another student), you should continue to show all the available meeting times.



Office Hours Sign Up

Student Name: Grace Student Email: gwu@gmu.edu Submit Clear

Email successfully sent from jdoe@gmu.edu

February, 2015						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2 ○7:00am ○7:30am ○8:00am	3 ○7:30am ○8:00am	4	5	6 ○1:00pm ○2:00pm	7
8	9 ○7:00am ○7:30am ○8:00am	10 ○7:30am ○8:00am	11	12	13 ○1:00pm 2:00pm -- Jon	14
15	16 ○7:00am ○7:30am ○8:00am	17 ○7:30am ○8:00am	18	19	20 ○1:00pm ○2:00pm	21
22	23 ○7:00am ○7:30am ○8:00am	24 ○7:30am ○8:00am	25	26	27 ○1:00pm ○2:00pm	28

You should continue to stick to the basics, using functions already available in PHP. Constants should be used for values that do not change, and variables for values that change during the course of execution. You should include Last Modified Date information on your page. This should be provided dynamically by using the built-in `date()` function within the `getlastmod()` function.

At the point of final submission, your application should be laid out in the *Lab Content* area of your web site. When laying out your web page, you should retain the common navigation and structural components (header and footer) of your website. This will require you to incorporate the SSI file references in this lab assignment as you did on the previous lab assignment. SSI files should not be moved, but rather you should provide the path to their original file location. The common navigation and structural components should present visual continuity across all the web pages of your website and should not shift at all when going from web page to web page. You should not update any SSI files (including the links within them) for this lab assignment. In addition, while the image does not demonstrate this you need to include a working link on this page to get back to your home page.

To be eligible for grading XHTML tables should not be used for the calendar or anywhere else in the assignment as using tables for positioning is not XHTML compliant. In order to create the output required of this assignment, you should use XHTML for the content of all files and CSS for the stylizing and layout. When setting up the positioning of the calendar and setup form, note that the layout should remain consistent and visible no matter the size of the browser window. Cascading styles necessary for this assignment should be implemented as part of the early submission points. While you should continue to use the external CSS you created for your home page, you should not modify or update the file. Your new CSS file should not conflict or modify the layout and positioning of earlier lab assignments.

For your application, you should use a file name that will automatically load if a user specifies only the folder name as part of the URL. The file should be uploaded to your Lab Assignment 2 directory. Your XHTML file should be well-formed and created using a strict XHTML DTD. You can create the web page using any text editor (Microsoft Notepad, RogSoft Notepad+, Adobe Homesite, etc) that is not WYSIWYG capable (Adobe Dreamweaver, Microsoft Word, etc).

As you are building your web page you should either submit that web page to the W3C Validator or use the Firefox HTML Validator to assure XHTML compliancy. Beyond these requirements, you should make sure that your website is visually appealing. In other words, once meeting the requirements listed, go ahead and be creative.

Approach
You should first make sure you meet the requirements of the first submission point before proceeding with the final requirements. While the accompanying grade sheet further details these requirements for the first submission point your code should generate a static calendar for the current month using control structures in a .php script file. This calendar can then be formatted using a new, external .css file. You can also begin to create the Office Hours Setup Form page with only the non-form code.

After the first submission point you can finish creating the Office Hours Setup Form. This is a simple XHTML file that will submit to the calendar .php file. Once the XHTML file is designed, work on passing these values from the XHTML file to the calendar .php file. Success is when the setup form page values display in the correct parts of the calendar.

After getting these two web pages to work together, work on the signup portion of the calendar .php file. This file will end up submitting entered/selected form values to itself. When creating hidden fields in the .php file for any values you need to retain from the Office Hours Setup Form, keep track of what the form fields were originally called on that setup form. This will allow the logic in the calendar page to continue to work as the field names will be the same. Finally add the emailing functionality, Last Modified Date, and any other required functionality.

Submission
You will need to upload these web/script pages and CSS file to your Lab Assignment 2 directory within your Zeus/Helios server space, while keeping the original SSI and CSS files in their original file location. Verify that all the uploaded pages work in the browser, as this is what will be used for grading. Also for each submission point, you must submit all the files you create to the appropriate Assignment drop box on Blackboard.

Notables

- Remember that the lab assignment must clearly display your name in the interface (if any) and be commented with your name in all files.
- Try to be consistent in the naming of your files as this will help you later on.
- Feel free to use your book and notes for general information on these technologies.
- If need be, for reference material on HTML, you may want to visit http://w3schools.com/html/ which we discussed in class.
- For reference material on CSS, you may want to visit http://w3schools.com/css/ which we discussed in class.
- For reference material on PHP, you may want to visit http://php.net which we discussed in class.
- Websites other than those provided in this lab should not be utilized.
- Just a reminder… this is individual work so while you can provide general assistance to other students while in the lab class, you should not work together or give solutions. Assistance provided to others while outside the lab class should only be through the use of the discussion forums.