

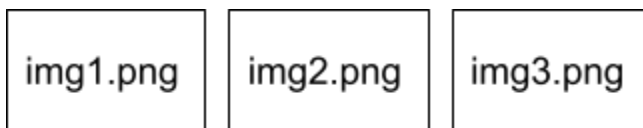
Level 1 (*Students will have to have created/filled the blogs table of the content DB*) - [DB Code](#)

1. Display a list of all the blog titles on a page, alphabetically.
2. Create variables of the necessary information for a blog entry. Use the variables to insert a new record into the blogs table.
3. Display all of the blog entries, ordered by descending date, in the following format. The title should be bold and all caps. The posted by information should be slightly smaller and in italics. On the next line is the text in normal font. Then, take the image list and for each image, draw a box with the image name in it. For an extra challenge, display only the Title and posted information and use a JQuery click event with the hide and show functions to show the full blog entry when the title is clicked.

TITLE (*posted by author on date*)

Text text text text. Text text text text. Text text text text. Text text text text.

Text text text text.



Level 2 (*Students will have to have created/filled the auditionees & scores table of the auditions DB*) - [DB Code](#)

1. Show an alphabetical list (last, first) of all students auditioning organized by voice part. In other words, all of the altos should be listed alphabetically, followed by an alphabetic list of bass, etc.
2. Truncate the data in the scores table. Then, create a page that takes information through GET variables for entering scores. This page should take the values of the variables and input the score information into the appropriate table. Additionally, this page should generate a drop down that lists all of the auditionees that have NOT been scored. Once a score has been submitted for a student, they should no longer appear in the drop down.
3. Create a reports page. The report page should give the user the option to see the list of auditionees and scores in order of priority. They should be given the following options: Music, Dance, Acting, Music & Acting, Music & Dance, Overall. Each option should produce a table of students (with voice part and role) and their scores, in order of the chosen scores. There should be a title at the top of the report denoting what the list is sorted by. The easiest way to do this will be to use buttons on a page that take you to a another page. Set the target url with a get variable attached that specifies the sort field.

Level 3 (*Students will have to have created/filled the teacher, parent & conference tables of the conferences DB*) - [DB Code](#)

1. Create a login script in a page. A person logs in by entering their email as their username and their last name as their password in get variables of the login.php url. Process the login to decide if the person is a parent, teacher or neither. If they are a parent, take them to a parent conferences home page. If they are a teacher, take them to a teacher conferences home page. Otherwise, redirect them to the login page and notify them that their credentials are not correct.
2. On the teacher page, show a list of conferences, in order. Each conference should show the date, time, parent name and student name to be discussed. The teacher id should be sent as a get variable to assist in querying the DB.
3. On the parent page, show that parent their conference schedule, in order. Be sure to show the date, time, teacher name (with title), room number and student being discussed. The parent id should be sent as a get variable to assist in querying the DB.
4. Create a page to schedule a new conference. The page should take all necessary values as get variables. When the page finishes, reload the parent page (of the conference) and show their updated schedule with the new

conference. ~~The form should submit to the same page, update the database and re-display the schedule with the new conference included. The form should show a list of teachers from the teacher table to choose from. For those that know JQuery, the form could start as hidden and show only when an “Add New Conference” button is clicked.~~

- ~~5. For an extra challenge, use session variables to maintain a user’s login status throughout their visit to the site. Then, when they want to leave, provide a logout button that destroys the session information.~~

1. Create a page that allows users to create announcements that are stored in an announcement table. The table should have the start date, end date, category and announcement. Then, write a page that shows the announcements, ordered by start date, organized by category.
2. Create a page that is set up to write email. Users should be prompted to enter the from, to, subject, etc. The date and time information should be automatically calculated/entered from the computer clock when the send button is pushed. You will need to figure out how to get the correct date/time through php. Then, store the information in a database. As for the output, when a user logs into the page, they should see any email that has been “sent” to them.
3. Create a page that loads in all of the databases available in MySQL. Users can choose one of the databases or to create a new one. Once they have selected/created a database. They should be able to name a table and choose how many fields that table should have. Then, the next page should be a page that helps them create their table(var names, type, primary key, etc).