

Karaoke Group Project

You will be working in groups of between three and five people to finish this project during class time. At the end, each group will show off their implementation in front of the class.

You are employed by a company that is hired by bars and other party venues to run their Karaoke nights. They want to make a new application that will allow people to use their phones to sign up for songs using a web interface. Your task is to design a database to store the required information, and to write the web applications for both the singer and the DJ.

The potential singer should be able to search through a list of songs by either the artist name, the title of the song, or the name of one of the contributors. A contributor is someone who has contributed to the creation of the song in some significant way, such as the author, the singer, the guitarist, the drummer, etc. There should be information on what contribution each of these contributors has made for each song. As an example, this should allow a user to find all of the songs that Paul McCartney has made contributions to, whether it was as a member of The Beatles or as a member of Wings.

Each song has at least one karaoke file, but can possibly have many. Each of these files will have a unique identifier. Since it may be important to the karaoke singer which version they are to perform, this identifier is what should be used when entering a song into the queue.

When searching for songs, it is possible that there may be many rows of results returned. When this happens, *the user should be able to click on a table column heading to sort the results based on the data value represented by that column*. The first click will sort the table in ascending order based on that column, then the next will sort in descending order, then back to ascending again, etc.

Upon choosing a version of a song, the user should be able to enter his song into one of two queues. One is a free for all, first come, first served queue that can be entered without charge. The other queue is accelerated, where the user can pay money to have his song get played earlier. You just need to allow people to sign up for the queues. The DJ will decide how to actually choose which song plays next at any given time.

You are also tasked with designing the DJ interface. This interface should show both queues, including relevant information about both the user and the version of the song they queued for. This information should include the user who requested the song, its title, the name of the band that performed it, and the special karaoke file ID associated with the version of the song that was selected. The free queue should be sorted based on the time the user signed up; first in, first out. The accelerated queue can be sorted in either time order or by the amount paid.

These are the goals your group should be working to achieve, in order:

1. Create an ER diagram that accurately models the database that will be needed to implement the application.
2. Once the ER diagram is done and approved by the professor, convert it into a relational schema (list of tables). Do not continue on to the SQL portion until this is done.
3. Create and run an SQL script that will create the tables from the relational schema in the step above.
4. Think up some sample data and insert it into your SQL tables. You should have enough where you can effectively test the application as you develop it.
5. Implement the web application using PHP. The following must be present, at a minimum:
 - a page that allows a user to specify an artist, song title, or contributor to search for
 - a page that shows the songs that match the criteria from the above page. Make sure to show all of the versions of each song found. The user should be able to choose a version of the song and enter into either of the queues. If he or she chooses the accelerated queue, find out from the user somehow how much they are willing to pay.
 - a page that shows the queues for the DJ. Using this page, the DJ should also be able to flag songs as already played once he has called the users up to perform. The already-performed songs will no longer show in the queue, but will remain present in the database.