# Assignment: Lab 3

You will finish the pages you created in lab 2, specifically salespeople.php, clients.php and calls.php. You will create a function called “display\_table” that will show the database records in tabular form upon page navigation. You will need to create appropriate prepared statements to make this happen. Here is one approach:





## Record Searching/Displaying

As indicated, each previously created page will need a paged table displayed on navigation. A reasonable default for records per page is 10, but this value should be defined in constants.php and not hard coded anywhere else. An excellent resource for page navigation UI can be found here: <https://getbootstrap.com/docs/4.0/components/pagination/>.

No matter which approach you use for page navigation, it will work by showing records:

(n – 1) x m + 1 thru n x m

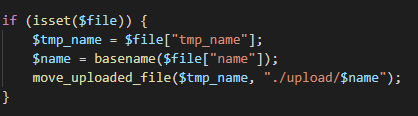
where n is the page number and m is the records per page constant. For instance, if n is page 2 and m is 10 records per page, you’ll be showing records (2 – 1) x 10 + 1 thru 2 x 10, ie 11 thru 20. Page 1 would be records 1 thru 10, page 3 would be record 21 thru 30, and so on. The paging component should only show pages that have records. For instance, if there are only 35 records, there should be no option to jump to page 5.

Note that the above formula is defined for the way humans count (ie starting at 1). You’ll need to make the appropriate adjustments to work with 0-based indexes of arrays.

The data should reflect the role of the signed in user. For instance, an administrator would be allowed to see all clients, while a salesperson would only be allowed to see his or her clients. Similarly, a salesperson would only see calls associated with his or her clients and would not be allowed to access the salesperson page at all.

## File Upload

The clients table will need to be updated to take the path of an uploaded file. This is the field called “logo\_path” as shown above. The corresponding prepared statements should reflect these changes. There is an excellent reference to the code needed for PHP to correctly receive the uploaded file found here: <https://www.php.net/manual/en/function.move-uploaded-file.php>. The relevant code will be:



“display\_table” will need to be adjusted to preview uploaded logos, rather than simply displaying the file location. One such approach might be to look for the label “logo\_path” and use appropriate logic for that.

File uploads will need basic validation based on file type and file size.

**The HTML input type for a file upload is cryptically called “file” (as opposed to “text”, “number” or “email”)**

## Password Update

You will create a new page called “change-password.php” that will only be accessible to signed-in users. That page will display a form asking for a password and another input to confirm it.



The form will submit to itself. If there’s a validation error, you will display the appropriate validation message. Passwords should be at least 3 characters in length. In a production system, we would expect the password rules to be much more restrictive.

You will create a new prepared statement called “user\_update\_password” that will hash the password and insert it into the correct database record. On successful password update, you will redirect the user to the dashboard, with an appropriate banner message.