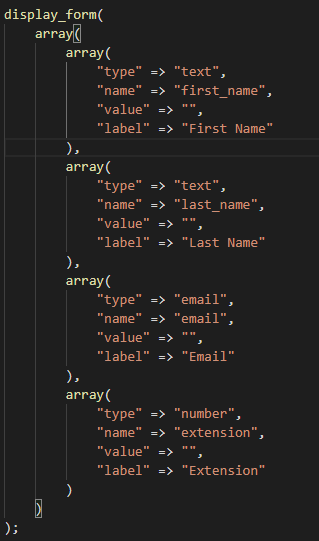
Assignment: Lab 2

You will create three new pages for your site: salespeople.php, clients.php and calls.php.

Each new page will use a new function that you will create called “display\_form” that will take, as its only parameter, an array of associative arrays.



Each item in the inner array corresponds to a field in the displayed form. For instance, in the above example, the third element will end up drawing an input of type “text”, name “email”, with no default value and a label that reads “Email”.

User Registration/Update

On navigation, the page located at /salespeople.php will check to see if the user is signed-in as an admin. If not, he or she will be redirected to the sign in page with an appropriate banner message. The form, like all forms on the site, will submit to itself. You are encouraged to style this form to match the rest of your site.

On submission (ie POST), the salesperson will be created. A banner message indicating success is required. If the form fails to validate, it will be displayed with fields prefilled with the appropriate information. Note that password information should never be “sticky”. Email addresses need to be checked at a basic level for validity (see email validation) and number fields should be checked to ensure they are numbers.

It is never sufficient to rely on forms for validation. These need to be checked at the server level. Invalid fields will not be sticky, and appropriate error messages will be displayed in the banner.

If the script fails to insert a record, there needs to be an error message, even if you can’t isolate the validation error.

The dynamic nav bar should include new links to the new pages, based on a user being logged in, and being logged in as an ADMIN or SALESPERSON

Form Inputs from Database Records

Similarly, clients (that is, customers of your company) can be created by admins or salespeople. If the user is a salesperson, on submit the entry will reflect that information. For instance, if I were to create a client, we can safely assume that it is my client and I am not creating him or her on someone else’s behalf. The data should reflect this.

If the script finds that the user is an admin, a dropdown (see information on select elements) should be created with a list of salespeople. The salesperson associated with the client, in this case, is the one selected by the admin. An admin will never directly be associated with a client.

Naturally, if the user has not signed in, the form will direct to the sign-in page instead.

Finally, the salesperson will be able to create a “call” record.

## Database Creation/Population

The new tables need to be created and seeded with default data. The clients should include a phone number, email, extension (optional), first name and last name. Note they will not require a password, since they are not users and will not be signing into the system. It is safe to assume that emails will be unique across the entire system but remember that emails make terrible database ids. It will also be necessary to associate every client with one and only one salesperson, using a foreign key (see foreign keys: <http://www.postgresql.org/docs/7.4/static/ddl-constraints.html#DDL-CONSTRAINTS-FK>)

You will need a way to track calls, including which client made the call, and the time of the call. Note that it is not necessary to explicitly state the salesperson who took the call, because we can safely assume it is the salesperson associated with the account. There is no need to store this information in the calls table.