Build yourself a simple CRM from scratch in PHP and MySQL

Richard

Feb 28, 2017 · 8 min read

Photo by Mike Kononov on Unsplash

Customer Relationship Management (CRM) is a system that manages customer interactions and data throughout the customer life-cycle between the customer and the company across different channels. In this tutorial, we are going to build a custom CRM in PHP, which a sales team can use to track customers through the entire sales cycle.

We'll be creating a simple CRM system for salespeople to:

- Access their tasks
- View their leads
- Create new tasks for each lead
- Create new opportunity
- Close a sale

Sales managers will be able to:

- Manage all customers
- Manage sales team
- View current sales activities

Download Demo Files

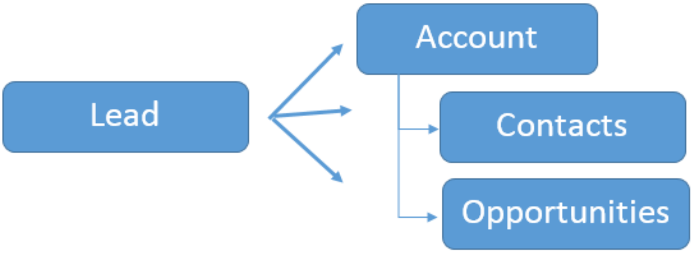
Building Blocks of a CRM

Here is a list of the essential components of the CRM:

- **Leads**: initial contacts
- **Accounts**: Information about the companies you do business with
- **Contact**: Information about the people you know and work with. Usually, one account has many contacts
- **Opportunities**: Qualified leads
- **Activities**: Tasks, meetings, phone calls, emails and any other activities that allow you to interact with customers
- **Sales**: Your sales team
- **Dashboard**: CRM dashboards are much more than just eye candy. They should deliver key information at a glance and provide links

to drill down for more details.

- **Login**: Salespeople and managers have different roles in the system. Managers have access to reports and sales pipeline information.



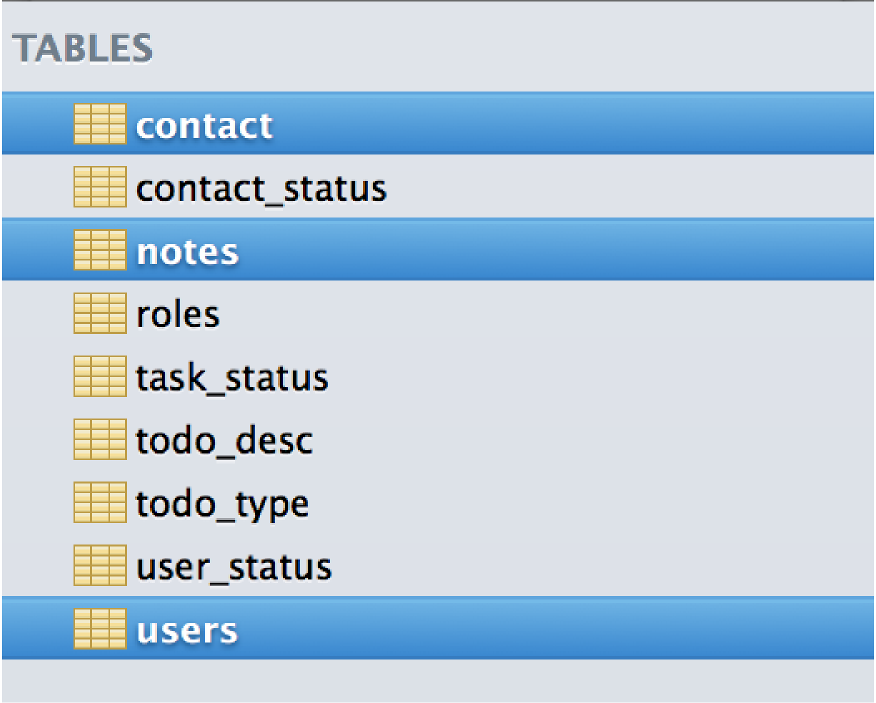
System Requirements

- PHP 5.3+,
- MySQL or MariaDB
- phpGrid

Create CRM Database

We will start by creating our custom CRM database. The main tables we will be using are:

- **contact**—contains basic customer data
- **notes**—holds information collection from Contact by sales people.
- **users**—information about sales people



TABLES

contact

contact_status

notes

roles

task_status

todo_desc

todo_type

user_status

users

The **Contact** table contains basic customer information including names, company addresses, project information, and so forth.

The **Notes** table stores all sales activity information such as meetings and phone calls.

The **Users** table holds login information about users of the system such as usernames and passwords. Users can also have roles, such as Sales or Manager.

All other tables are lookup tables to join to the three main relational database tables.

- **contact_status**—contains contact status such as Lead and Opportunity. Each indicates a different stage in a typical sales cycle
- **task_status**—the task status can be either Pending or Completed

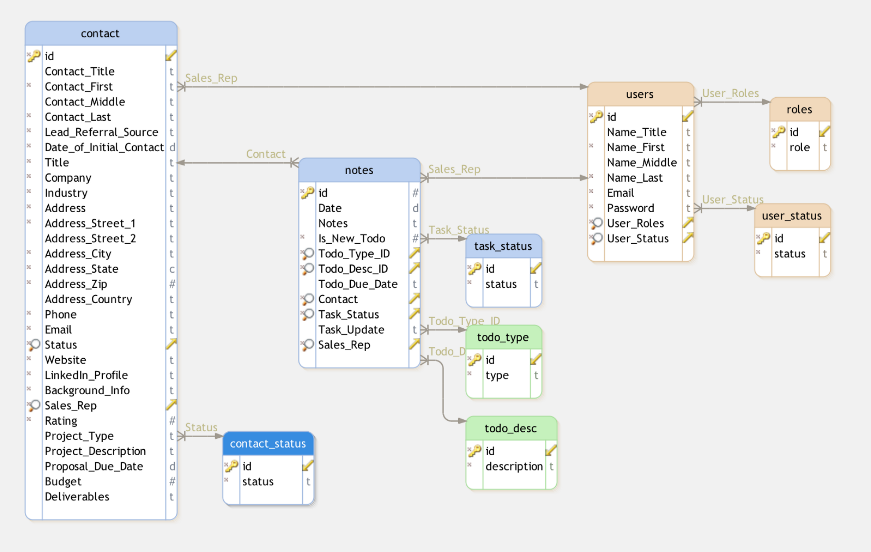
- **user_status**—a sale person can be Active or Inactive
- **todo_type**—a type of task either Task or Meeting
- **todo_desc**—description of a task such as Follow Up Email, Phone Call, and Conference etc.
- **roles**—a user can be either a Sales Rep or a Manager

Complete Database Schema Diagram

A database schema is the structure that represents the logical view such as tables, views, or primary and foreign keys of the entire database. A database schema includes entities and the relationship among them.

It is a good practice to have one primary key for each table in a relational database. A primary key is a unique identifier for each record. It can be the social security number (SSN), vehicle identification number (VIN), or auto-increment number. This is a unique number that is generated when a new record is inserted into a table.

Below is the database diagram of our simple CRM. The key symbol in each table represents the table primary key. The magnifying glass indicates foreign key linking another table in the database. Sometimes we call it the “lookup” table.



install.sql

Once you have an understanding of the database table structure, find the “install.sql” script in the `db` folder and use a MySQL tool such as MySQL Workbench or Sequel Pro to run the SQL script. It should create a new relational database named `custom_crm` and its database tables.

A Side Note on ZenBase

```
The CRM application is also one of the many application
templates readily available at ZenBase (built on the top of
phpGrid) for anyone — with or without coding skills — to use
and customize for their own needs.
```

Setup phpGrid

Our CRM contains many datagrids. The datagrid is a spreadsheet-like data table that displays rows and columns representing records and fields from the database table. The datagrid gives the end-user ability to read and write to database tables on a webpage. We can use a datagrid tool from phpGrid. We use a tool instead of building them from scratch because developing the datagrid is usually tedious and error-prone. The datagrid library will handle all internal database **CRUD (Create, Remove, Update, and Delete)** operations for us with better and faster results with little code. To install phpGrid, follow these steps:

1. Unzip the phpGrid download file.
2. Upload the phpGrid folder to the `phpGrid` folder.
3. Complete the installation by configuring the `conf.php` file.

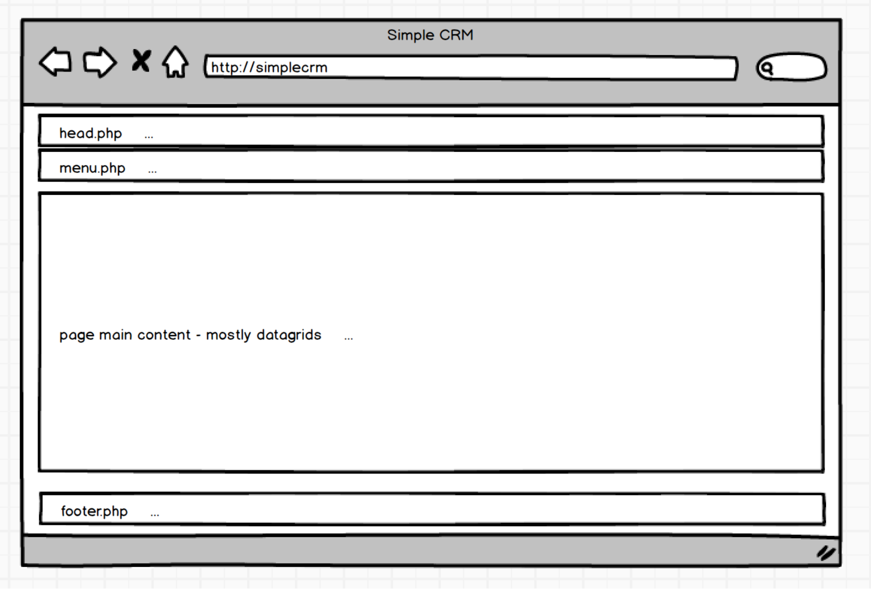
Before we begin coding, we must specify the database information in `conf.php` , the phpGrid configuration file. Here is an example of database connection settings:

```
<?php
define('PHPGRID_DB_HOSTNAME', 'localhost');
define('PHPGRID_DB_USERNAME', 'root');
define('PHPGRID_DB_PASSWORD', '');
define('PHPGRID_DB_NAME', 'custom_crm');
define('PHPGRID_DB_TYPE', 'mysql');
```

- **PHPGRID_DB_HOSTNAME**—web server IP or host name
- **PHPGRID_DB_USERNAME**—database user name
- **PHPGRID_DB_PASSWORD**—database password
- **PHPGRID_DB_NAME**—database name of our CRM
- **PHPGRID_DB_TYPE**—type of database
- **PHPGRID_DB_CHARSET**—always ‘utf8’ in MySQL

To learn more about configuring phpGrid, check out phpGrid complete installation guide.

Page Template



Before we can start building our first page of the CRM, it is a good practice to make the reusable page items such as header and footer.

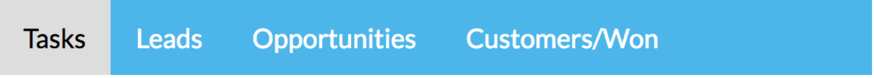
The page will comprise of a header, menu, body and footer. We will start by creating a reusable page template.

head.php

This is a basic HTML5 template header. It includes a link to a custom stylesheet that will be created in a later step.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lan
<head>
<title>My Custom CRM</title>
<meta http-equiv="Content-Type" content="text/html; charset=
<link type="text/css" rel="Stylesheet" href="../css/style.cs
```

menu.php



```
<div id="menu">
<ul>
<li><a href="tasks.php" <?php if($_GET['currentPage'] == 'ta
<li><a href="leads.php" <?php if($_GET['currentPage'] == 'le
<li><a href="opportunities.php" <?php if($_GET['currentPage'
<li><a href="customerwon.php" <?php if($_GET['currentPage']
</ul>
```

Notice the usage of `$_GET['currentPage']` . Each page will set a value which will highlight the name of the current page on the top menu bar.

Include the following code in style.css for menu styling (minified). It will transform the above, unordered list into a menu.

```
#menu ul { list-style-type: none; margin: 0; padding: 0; ove
```

footer.php

Simple closing body and html tags.

```
</body>
</html>
```

Complete Page Template

This is the complete page template. The main content will go after `Section Title` .

```
<?php
use phpGrid\C_DataGrid;
require_once("../phpGrid/conf.php");
include_once('../inc/head.php');
?>
<h1>My Custom CRM</h1>
<?php include_once('../inc/menu.php'); ?>
<h3>Section Title</h3>
<?php
```

CRM Main Pages

Are you still with me? Good! We can now finally develop the first page in our CRM.

Our CRM for the sales team members has four pages:

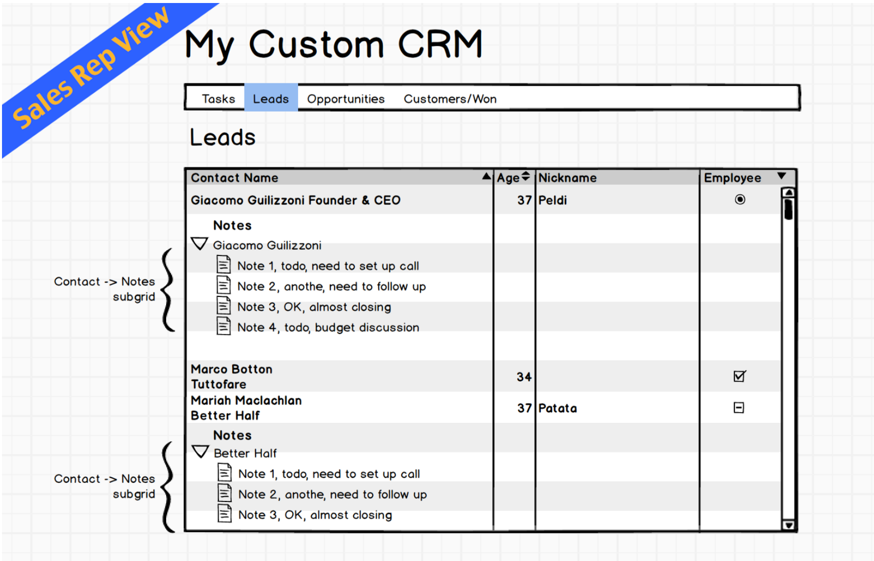
- **Tasks**
- **Leads**

- **Opportunities**
- **Customers/Won**

Each page indicates a different stage in a typical sales cycle.

Sale People Page Design Mockup

Here’s our CRM design mockup for the sales people.



Contact Name ▲	Age ⬍	Nickname	Employee ▼
Giacomo Guilizzoni Founder & CEO	37	Peldi	◉
Notes			
▽ Giacomo Guilizzoni			
Note 1, todo, need to set up call			
Note 2, anothe, need to follow up			
Note 3, OK, almost closing			
Note 4, todo, budget discussion			
Marco Botton Tuttofare	34		☑
Mariah Maclachlan Better Half	37	Patata	⊟
Notes			
▽ Better Half			
Note 1, todo, need to set up call			
Note 2, anothe, need to follow up			
Note 3, OK, almost closing			

Tasks Page

When a sales team member logged in, the first page he sees is a list of current tasks.

As you may recall, our Notes table holds all the sales activity information. We can create a datagrid and populate it from the Notes table using phpGrid.

The Tasks page main content is a datagrid. The following two lines will give us a list of tasks of the current sales person.

```
<?php
$dg = new C_DataGrid("SELECT * FROM notes", "ID", "notes");
$dg -> display();
?>
```

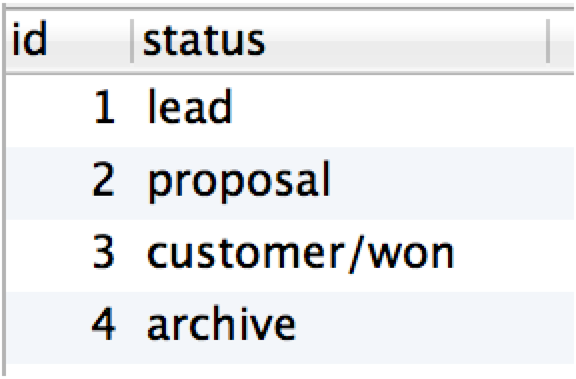
- The first line creates a phpGrid object by passing the SELECT SQL statement, its primary key— `ID` , and then the name of the database table - `notes` .
- The second and the final line calls display() function to render the datagrid on the screen. Check out the basic datagrid demo for more detail.

Leads Page

The leads page contains list of current leads that the sales person is responsible for. Each Lead can have one or many Notes. We will use the phpGrid master-detail feature for that.

We also need to use set_query_filter() to display only the leads, `Status = 1` , and only for the current sales person.

Contact status table



id	status
1	lead
2	proposal
3	customer/won
4	archive

```
<?php
$dg = new C_DataGrid("SELECT * FROM contact", "id", "contac
$dg->set_query_filter(" Status = 1 && sales_rep = 1 ");
// the detail grid displays notes about a lead
$sdg = new C_DataGrid("SELECT * FROM notes", "id", "notes")
$sdg->set_query_filter(" Sales_Rep = 1 ");
$sdg->enable_edit();
```

Opportunities Page

A Lead becomes an Opportunity once it is qualified. The Opportunities page is similar to the Leads page. The only difference is the filtered status code in set_query_filter is `Status = 2` .

```
<?php
$dg = new C_DataGrid("SELECT * FROM contact", "id", "contact
$dg->set_query_filter(" Status = 2 && sales_rep = 1 ");
$sdg = new C_DataGrid("SELECT * FROM notes", "id", "notes");
$sdg->set_query_filter(" Sales_Rep = 1 ");
$sdg->enable_edit();
$dg->set_masterdetail($sdg, 'Contact', 'id');
```

Customers/Won Page

Customers/Won has the `Status = 3` . Similar to Leads and Opportunities, Customers/Won can also have Notes.

```
<?php
$dg = new C_DataGrid("SELECT * FROM contact", "id", "contact
$dg->set_query_filter(" Status = 3 && sales_rep = 1 ");
$sdg = new C_DataGrid("SELECT * FROM notes", "id", "notes");
$sdg->set_query_filter(" Sales_Rep = 1 ");
$sdg->enable_edit();
$dg->set_masterdetail($sdg, 'Contact', 'id');
```

That’s all there is to it for sales people in our simple CRM.

Manager Dashboard

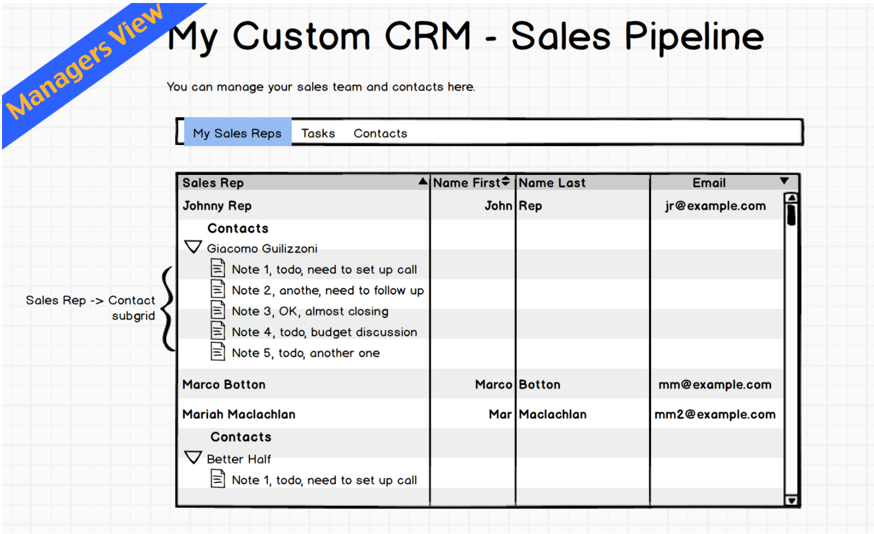
Photo by Eaters Collective on Unsplash

The sales manager will have access to all records in the sales pipeline as well as the ability to manage sales team and customer data.

We will have a single web page with tabbed menu similar to the phpGrid tabbed grid demo.

Manager Dashboard Design Mockup

My Sales Reps



My Custom CRM - Sales Pipeline

You can manage your sales team and contacts here.

Sales Rep	Name First	Name Last	Email
Johnny Rep	John	Rep	jr@example.com
Contacts			
Giacomo Guilizzoni			
Note 1, todo, need to set up call			
Note 2, anothe, need to follow up			
Note 3, OK, almost closing			
Note 4, todo, budget discussion			
Note 5, todo, another one			
Marco Botton	Marco	Botton	mm@example.com
Mariah Maclachlan	Mar	Maclachlan	mm2@example.com
Contacts			
Better Half			
Note 1, todo, need to set up call			

Main content

Each tab represents a table in the CRM database. `$_GET['gn']` will store the table name. It dynamically generates the datagrid based on table name passed.

```
<?php
$tableName = (isset($_GET['gn']) && isset($_GET['gn']) !==
switch($tableName){
        case "users":
                $dg = new C_DataGrid("SELECT * FROM users",
                $sdg = new C_DataGrid("SELECT * FROM contac
                $dg->set_masterdetail($sdg, 'sales_rep', 'i
                break;
        case "notes":
                $dg = new C_DataGrid("SELECT * FROM notes",
                break;
        case "contact":
                $dg = new C_DataGrid("SELECT * FROM contact
                $sdg = new C_DataGrid("SELECT * FROM notes"
                $sdg->enable_edit();
```

It's very easy to integrate jQueryUI Tabs with phpGrid. Please refer to the phpGrid Tabbed Grid demo for more information.

My Sales Rep Page

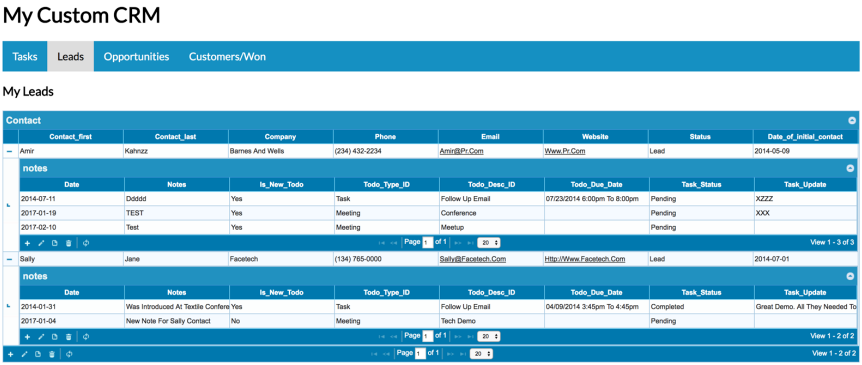
Since a sales manager needs to quickly find out whom a sale person is working with, we added a detail grid `$sdg` populated from contact table and link with the master grid.

```
$sdg = new C_DataGrid("SELECT * FROM contact", "id", "contac
$dg->set_masterdetail($sdg, 'sales_rep', 'id');
```

`sales_rep` is the connecting key in `contact` table to the `id` that is the foreign key in `users` table. Remember the `users` stores all of our sales people information.

Screenshots

CRM — Sales Screen



My Custom CRM

My Leads

Contact

	Contact_first	Contact_last	Company	Phone	Email	Website	Status	Date_of_initial_contact
–	Amir	Kahnzz	Barnes And Wells	(234) 432-2234	Amir@Pr.Com	Www.Pr.Com	Lead	2014-05-09

notes

Date	Notes	Is_New_Todo	Todo_Type_ID	Todo_Desc_ID	Todo_Due_Date	Task_Status	Task_Update
2014-07-11	Ddddd	Yes	Task	Follow Up Email	07/23/2014 6:00pm To 8:00pm	Pending	XZZZ
2017-01-19	TEST	Yes	Meeting	Conference		Pending	XXX
2017-02-10	Test	Yes	Meeting	Meetup		Pending	

Page 1 of 1 20 View 1 - 3 of 3

	Contact_first	Contact_last	Company	Phone	Email	Website	Status	Date_of_initial_contact
–	Sally	Jane	Facetech	(134) 765-0000	Sally@Facetech.Com	Http://Www.Facetech.Com	Lead	2014-07-01

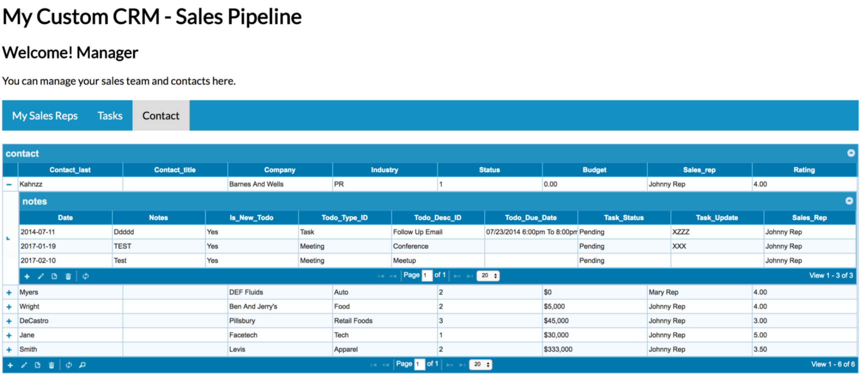
notes

Date	Notes	Is_New_Todo	Todo_Type_ID	Todo_Desc_ID	Todo_Due_Date	Task_Status	Task_Update
2014-01-31	Was Introduced At Textile Confere	Yes	Task	Follow Up Email	04/09/2014 3:45pm To 4:45pm	Completed	Great Demo. All They Needed To
2017-01-04	New Note For Sally Contact	No	Meeting	Tech Demo		Pending	

Page 1 of 1 20 View 1 - 2 of 2

Page 1 of 1 20 View 1 - 2 of 2

CRM — Manager Screen



My Custom CRM - Sales Pipeline

Welcome! Manager

You can manage your sales team and contacts here.

contact

	Contact_last	Contact_title	Company	Industry	Status	Budget	Sales_rep	Rating
–	Kahnzz		Barnes And Wells	PR	1	0.00	Johnny Rep	4.00

notes

Date	Notes	Is_New_Todo	Todo_Type_ID	Todo_Desc_ID	Todo_Due_Date	Task_Status	Task_Update	Sales_Rep
2014-07-11	Ddddd	Yes	Task	Follow Up Email	07/23/2014 6:00pm To 8:00pm	Pending	XZZZ	Johnny Rep
2017-01-19	TEST	Yes	Meeting	Conference		Pending	XXX	Johnny Rep
2017-02-10	Test	Yes	Meeting	Meetup		Pending		Johnny Rep

Page 1 of 1 20 View 1 - 3 of 3

	Contact_last	Contact_title	Company	Industry	Status	Budget	Sales_rep	Rating
+	Myers		DEF Fluids	Auto	2	$0	Mary Rep	4.00
+	Wright		Ben And Jerry's	Food	2	$5,000	Johnny Rep	4.00
+	DeCastro		Pillsbury	Retail Foods	3	$45,000	Johnny Rep	3.00
+	Jane		Facetech	Tech	1	$30,000	Johnny Rep	5.00
+	Smith		Levis	Apparel	2	$333,000	Johnny Rep	3.50

Page 1 of 1 20 View 1 - 6 of 6

Live Demo

CRM Sales Rep Screen | CRM Managers screen

Need to Write Even Less Code?

If you are new to programming and are not yet comfortable with coding, you may want to check out **ZenBase** that is built on the top of the phpGrid. The CRM is but one of the many application templates readily available at ZenBase for anyone—*with or without* coding skills—to use and customize for their own needs.

Complete Source Code on GitHub

phpcontrols/phpgrid-custom-crm

phpgrid-custom-crm - Custom CRM Demo - Learn to build yourself a custom CRM in PHP and...

github.com

. . .

Thanks for reading. If you enjoyed this article, please hit that clap button 👏 to help others find it and follow me on Twitter.