

MAGNET

tangible customized solutions
for your e-business

PHP-GTK

Endless Possibilities

a paper by

Nirav Mehta
Vaishali Master
& Piyush Shah

Executive Summary

PHP is most widely used as a web application programming language. It obviously has many advantages in that area, but as a language, PHP is a general purpose programming language. What if you could develop desktop applications using PHP?

PHP-GTK is an extension to PHP that allows you to develop standalone GUI applications. These applications can run on multiple platforms. You can utilize the power of PHP and use graphical desktop application like interface, this fact opens doors to endless possibilities. You can develop many applications with the existing PHP-GTK framework, and as it progresses, we see a good possibility of seeing PHP everywhere from the web to the wireless!

This paper is an introduction to PHP-GTK and tells you why you should be excited about it! We also talk about a few sample applications, including our PHPGale, which is a proof of concept script to fetch data from a source and display it in a desktop application.

Introduction

What is GTK+?

GTK+ stands for GIMP Tool Kit. GIMP (GNU Image Manipulation Program) is widely popular image manipulation software for Linux. GTK+ has grown from GIMP and is now the core of Gnome, a desktop environment like Windows. There are hundreds of applications built on GTK+ today. GTK+ has also been ported to Windows and BeOS.

GTK+ is actually a collection of libraries that provide you object oriented framework to develop graphical user interfaces. These libraries are GLib, GDK and GTK.

What is PHP?

PHP is the most popular module with the world's most popular web server software Apache. PHP is primarily used to develop web based applications. PHP has been successfully used to develop things ranging from simple form submissions to groupware and e-commerce applications.

PHP is all the more popular because of its support for many databases and the large range of extensions available. These extensions provide support for even things like .Net and Java!

PHP is open source, and if you think PHP does not support something you like, you are free to write up your own extension!

Now you can develop cross platform, database aware standalone GUI applications in your favorite scripting language - PHP!

What is PHP-GTK?

PHP-GTK is an extension to PHP that allows you to develop standalone GUI applications. PHP-GTK is cross platform, which means your applications will run on Linux and Windows without any change! PHP-GTK provides you a framework to develop window based GUI applications windows, buttons, drop downs et al not in a browser, but as an application itself!

This is tremendous push for PHP, and we expect PHP-GTK to be very popular. In the following pages, you will know what it is all about, and what can it do.

Basics of PHP-GTK

PHP-GTK is an object oriented framework. You create an instance of a GTK class and then call the methods or properties on it. If you have done object oriented programming in PHP, it should be easy to get started. Different actions are taken on different signals, which are raised by certain events.

Widgets

Widget is a term used for any user interface. A widget can be a text box, label, frame, window or any other GUI component. GTK widgets are all derived from the abstract base class `GtkWidget`, and inherit the methods, signals and properties implemented therein.

From a programming point of view a widget has five parts to its life cycle:

Creation

In PHP-GTK this is done by creating an object, e.g. `$window = &new GtkWidget();`

Placement

This is the step of adding a widget to a container. This is very straightforward. The syntax is `$container->add($widget);`

Signal Connection

This is the step of setting up callback functions, functions that will be executed on a particular event.

e.g. `$widget->connect("event", "my_focus_func");`

Here, "event" is a predefined state such as "clicked" and "my_focus_func" is the called subroutine.

Display

You can control the visibility of a widget using the display functions. Display is started by calling `$widget->show();` and finished by `$widget->hide();`

PHP-GTK provides an object oriented framework to create windowed applications.

Widgets are the core components of PHP-GTK. You create a widget, place it, associate it with an action and display it! That's as simple as it sounds!

Destruction

Widget destruction is automatically done, when the `gtk::main_quit()` function is called.

Containers

Consider a container as a widget that can hold another widget. `GtkWindow`, `GtkTable` and `GtkBox` are containers that immediately come to mind. Apart from this, they are just like any other widget. Containers are derived from the `GtkContainer` base class, which itself is derived from the `GtkWidget`.

Signals

Signals are notifications emitted by widgets.

Signals are GTK+ equivalent of events in Windows. While programming Graphical User Interfaces (GUI), it is often necessary to respond to actions performed by the user or initiated within the program itself. Signals are used to let the program know that something happened. This might be, for example, a user clicking on a `GtkButton`, or a change being made to a `GtkAdjustment` value.

Signals are inherited by objects, just as methods are. A widget can emit any signals that its ancestor objects have defined, as well as its own specific signals.

Callbacks

Callbacks are functions that are invoked when signals are emitted by widgets.

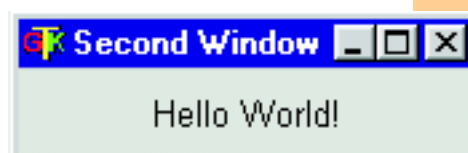
You specify the function that should be called by connecting the function to the signal. The callback is also known as the signal handler function. This can be either the widget's default handler or a user-defined handler, that is, a function written by the programmer.

Sample Applications

Hello World!

Hello World! The first program you will see in any language. Here's how you can do "Hello World!" in PHP-GTK!

Alright, let's do something more than Hello World! This code actually creates a window with a button and associates a callback function with the "clicked" event for this button. Here are the screenshots!



Signals are raised on a certain event, and you perform action by associating a callback function for a signal.

You can either use the default callback function or write one of your own. And it is just like a normal PHP function. Use the data and put the business logic.

```
<?php
// Load the GTK library, based on the operating system
if (!class_exists('gtk'))
{
    if (strtoupper(substr(PHP_OS, 0, 3)) == 'WIN')
        dl('php_gtk.dll');
    else
        dl('php_gtk.so');
}

// What to do when the window is closed?
function shutdown()
{
    print("Shutting down...\n");
    gtk::main_quit();
}

// Ok, this is the action of the clicked event on the
// first window button
function you_clicked($button)
{
    // Create a new window
    $window = &new GtkWindow();
    // Now set some properties and create a label
    $window->set_title("Second Window");
    $label = &new GtkLabel("Hello World!");
    // Add the label to our window!
    $window->add($label);
    // If this is destroyed, call
    // the shutdown function
    $window->connect("destroy", "shutdown");
    // Display all the widgets
    $window->show_all();
    return false;
}

// The first window!
$window = &new GtkWindow();
$window->set_title("First Window");
$window->connect("destroy", "shutdown");
$window->set_border_width(10);

$button = &new GtkButton("Click Me!!");
$button->connect("clicked", "you_clicked");
$window->add($button);

$window->show_all();

// All set, call the main method to show up
// the interface
gtk::main();

?>
```

Load the PHP-GTK extension, create widgets, associate action, and we have our Hello World application ready.

It's all Object Oriented. So you can set the properties or call methods of a GTK widget with the `->` operator.

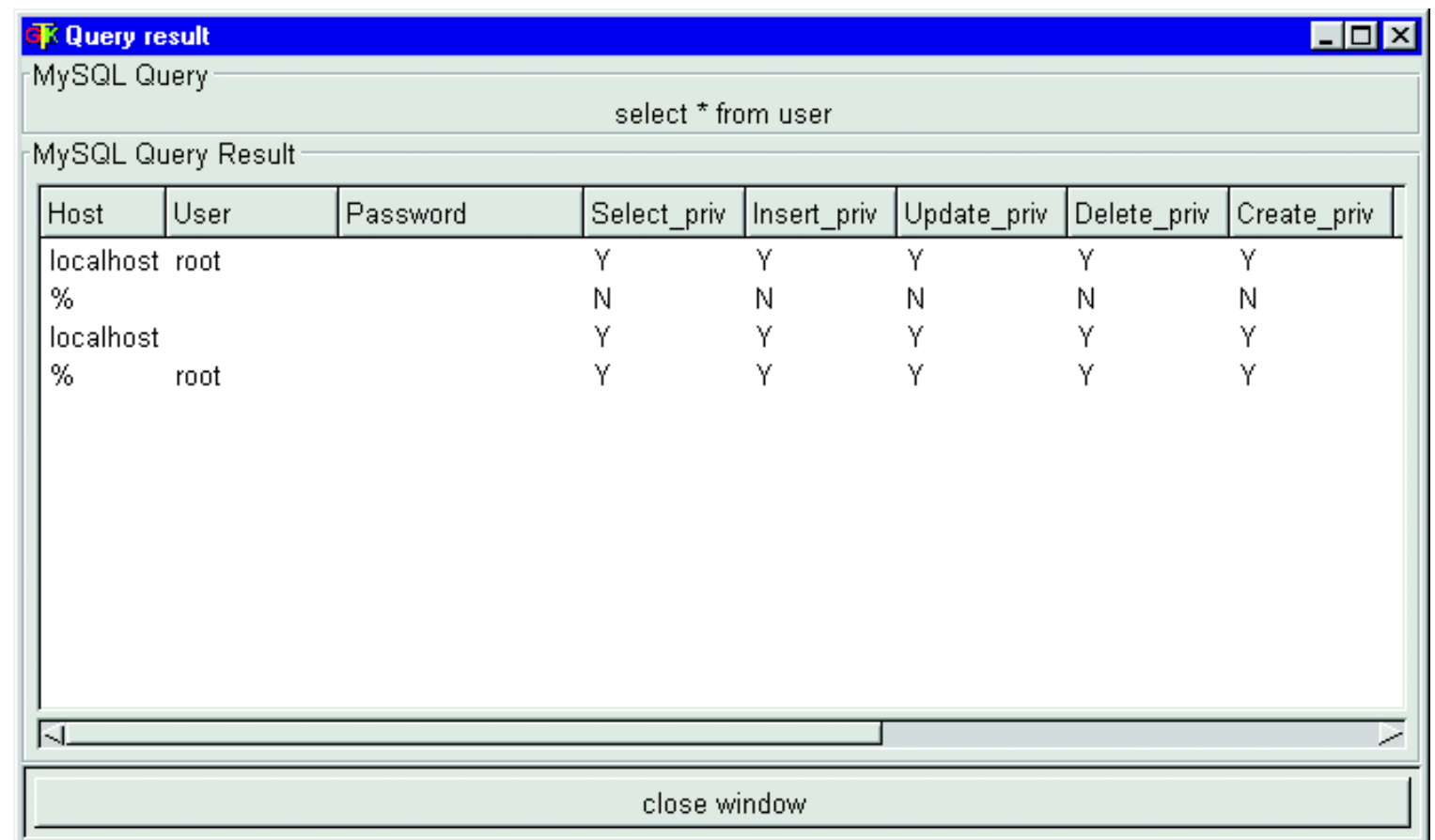
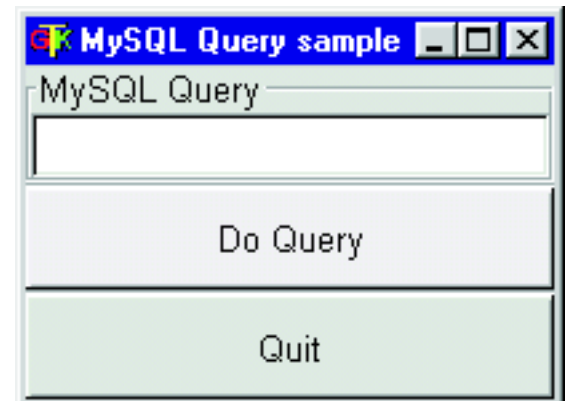
Go through the code, and things should be self explanatory! Doing simple things is always easy! On instructions on how to download and setup PHP-GTK refer to the resources section.

MySQL Query Tool

Here's one more, and rather complex example of what you can do with PHP-GTK. This is a GUI application that lets you query the MySQL database. You enter the query, the script will execute it and show you the results in a window!

Let's see the screenshots first, and then the code!

This is the first window that comes up when you run the script. Enter any query here and you will be shown the results in a new window that pops up like below!



Here's the code:

```
<?
/*
A sample script that takes SQL query from user,
Sends it to the MySQL server and displays the result of query in new window.
DON'T FORGET TO CHANGE SERVER, USERNAME AND PASSWORD IN MYSQL_PCONNECT
FUNCTION!! AND DATABASE IN DB_QUERY FUNCTION. MAKE SURE MYSQL SERVER IS UP!
I'll be glad for any comments.
Adam Rambousek - rambousek@volny.cz
*/
```

```
if (strtoupper(substr(PHP_OS, 0, 3)) == 'WIN')
    dl('php_gtk.dll');
else
    dl('php_gtk.so');

$windows = array();

function delete_event($window, $event)
{
    $window->hide();
    return true;
}

function close_window($widget)
{
    $window = $widget->get_toplevel();
    $window->hide();
}

/*
 Called when clist column is clicked. It sets sorting by the clicked column.
*/
function clist_click_column($clist, $column) {
    $clist->set_sort_column($column);
    $clist->sort();
}

/*
 Function displaying the result of query.
*/
function do_query($query)
{
    global $windows;

    //if the query_window is opened, let's close it
    if (isset($windows['query_window'])) {
        close_window($windows['query_window']);
    }

    $window = &new GtkWindow;
    $windows['query_window'] = $window;
    $window->set_name('query_window');
    $window->connect('delete-event', 'delete_event');
    $window->set_policy(false, true, false);
    $window->set_title('Query result');
    $window->set_ufposition(220, 85);

    $box1 = &new GtkVBox();
    $window->add($box1);

    //frame displaying entered sql query
    $frame = &new GtkFrame('MySQL Query');
```

```
$box1->pack_start($frame,false);
$label = &new GtkLabel($query->get_text());
$frame->add($label);

//frame displaying clist with the query result
$frame = &new GtkFrame('MySQL Query Result');
$box1->pack_start($frame,true);

//we'll display the result with scrollbars
$scrolled_win = &new GtkScrolledWindow();
$scrolled_win->set_border_width(5);
$scrolled_win->set_policy(GTK_POLICY_AUTOMATIC, GTK_POLICY_AUTOMATIC);
$frame->add($scrolled_win);

/*here we deal with the result. If mysql_query called with the
query entered in main window, return a result we can display clist
*/
if ($result = mysql_query($query->get_text())) {
    /*
    at first, keys array contains the names of columns
    */
    if ($data = mysql_fetch_array($result)) {
        $i=0;
        $keys = array();
        while (list($key,$val) = each($data)) {
            if ($i%2) $keys[]=$key;
            $i++;
        }
    }

    /*
    now we can prepare the clist, keys are the titles of columns and
    the number of columns is equal to the number of keys
    */
    $clist = &new GtkCList(count($keys), $keys);
    $clist->connect('click_column', 'clist_click_column');

    //we sets the auto_resize for each column
    for ($i=0;$i<count($keys);$i++)
        $clist->set_column_auto_resize($i, true);
    $scrolled_win->add($clist);

    /*
    now the data from result
    we get the data from each row to row_data array and then we
    append this array to the clist as a new row
    */
    do {
        for($i=0; $i < count($data)/2; $i++) {
            $row_data[$i] = $data[$i];
        }
        $clist->append($row_data);
    }while ($data = mysql_fetch_array($result));
}
```

```
}

$button = &new GtkButton('close window');
$button->connect('clicked', 'close_window');
$box1->pack_start($button,false);
$button->set_flags(GTK_CAN_DEFAULT);
$button->grab_default();
$window->show_all();
}

function main_window()
{
    $window = &new GtkWindow();
    $window->set_policy(false,true,false);
    $window->set_name('main_window');
    $window->set_title('MySQL Query sample');
    $window->set_uposition(80,80);
    $window->connect_object('destroy', array('gtk', 'main_quit'));
    $window->connect_object('delete-event', array('gtk', 'false'));

    $box1 = &new GtkVBox();
    $window->add($box1);

    $frame = &new GtkFrame('MySQL Query');
    $box1->pack_start($frame,false);

    $entry = &new GtkEntry();
    $frame->add($entry);

    $separator = &new GtkHSeparator();
    $box1->pack_start($separator,false);

    $button = &new GtkButton('Do Query');
    $button->connect_object('clicked', 'do_query',$entry);
    $box1->add($button);

    $separator = &new GtkHSeparator();
    $box1->pack_start($separator,false);

    $button = &new GtkButton('Quit');
    $button->connect_object('clicked', array('gtk', 'main_quit'));
    $box1->add($button);

    $window->show_all();
}

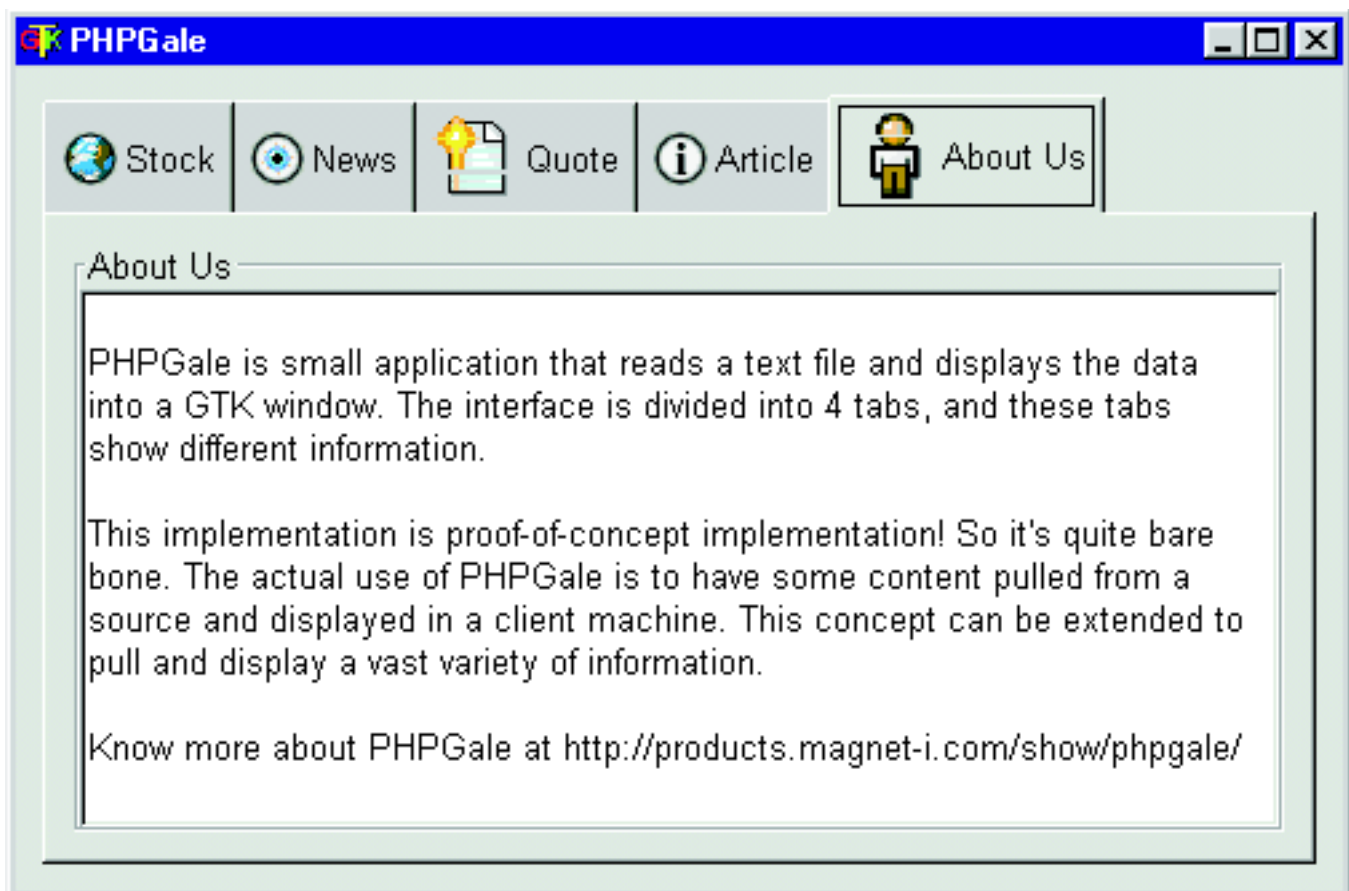
$connect = mysql_pconnect("SERVER","USERNAME","PASSWORD") or die("can't
connect to server");
mysql_db_query("DATABASE",$connect);
main_window();
Gtk::main();
?>
```

PHPGale - Staying on the cutting edge!

What is it?

PHPGale is small application that reads a text file and displays the data into a GTK window. The interface is divided into 5 tabs, and these tabs show different information.

This implementation is proof-of-concept implementation! So it's quite bare bone. The actual use of PHPGale is to have some content pulled from a source and displayed in a client machine. This concept can be extended to pull and display a vast variety of information.



Applications

PHPGale can be installed on a desktop and it can pull data from a server. This server could be an intranet or internet server. You can use PHPGale for announcements, news, stock tickers and anything that can operate on pulling data from a server and displaying it locally.

Download

Download the latest version of PHPGale from the following URL:

<http://products.magnet-i.com/show/phpgale/>

Future plan

Here are the things on our list for PHPGale.

Convert the input file into an XML file, so that data exchange becomes easy.

Content display logic to be separated from application logic

Design control over the display of the content

Automatically starting the application at a fixed time everyday, using a cron / scheduler activity

Image support, creating graphs on the fly from the incoming XML data

We also invite your comments on the project!

What lies ahead?

PHP-GTK puts a good equation about desktop applications in favor of PHP. Based on our experience in PHP, and the research in GTK, we believe there is a very good future for PHP-GTK. PHP is growing very fast, and PHP-GTK too is under constant development.

As of now, there are only 4 or 5 applications done in PHP-GTK. We foresee about a dozen very good applications coming out in PHP-GTK in the next 6 months. There will be many others on a smaller scale, and applications which are not open source. Organizations will use this technology for building tools linked to their websites. Things like an offline management tool or a mailing list manager could be a good deal.

Provided the vast amount of applications already available in GTK+ on Linux and Windows, a lot of experimental tools will come forward. There are millions of Linux and PHP enthusiasts world over, many of them are excited about PHP-GTK, and quite a few will contribute applications to the open source world.

You can think of just about any application you see around, and there are good chances that you can do that with PHP-GTK. Although, it would not yet provided you advanced features or drag and drop development flexibility, there would soon be libraries available, which you can use easily in your own applications.

Looking a little still further in the future, we have no reason to disbelieve why PHP-GTK can not be placed with Java as regards desktop application development. PHP is increasingly adopting features from Java, and keeping PHP's flexibility and ease of use, PHP-GTK can provide a very good option to develop relatively simple applications.

There is also a bright future, if GTK gets ported to palmtop systems. Then you can build just about anything, and it would work, just about anywhere!

If GTK+ gets ported to Palmtops, you could see PHP-GTK just about everywhere! From web to wireless!

We foresee about a dozen very good PHP-GTK applications coming up in the next 6 months. There will be hundreds of smaller ones, and a lot of non-open source solutions.

Resources

PHP

Learn more about what PHP is, and explore the possibilities!
<http://www.php.net/>

The company that brought PHP4 Zend.
<http://www.zend.com/>

Find out some great open source PHP applications:
<http://www.hotscripts.com/>

PHP-GTK

The official site for PHP-GTK. - <http://gtk.php.net/>

PHP-GTK Manual - <http://gtk.php.net/manual/en>

Tutorials / Manual - <http://gtk.miester.org>

Official GTK site - <http://www.gtk.org>

Download / Install

You can download the latest version of PHP and PHP-GTK from
<http://www.php.net> and <http://gtk.php.net>

We suggest you refer to detailed installation guide provided on these sites. A simple installation on Linux or Windows should not be a problem.

If you are using PHP-GTK on windows, we suggest you decide an extension for this (e.g. .phpg or .php4), and associate it with the PHP executable you get from the GTK installation.

The official PHP-GTK site is a great starting point for a beginner. Refer to the GTK+ sites for in-depth coverage of the topic including APIs.

Installing PHP-GTK is very simple. Installing it on Windows is even simpler!

About MagNet

MagNet WebPublishing is a technology driven web application development and design firm. We are a privately-held and headquartered in India. Our programming expertise, prompt service and strong client support combined with competitive pricing have allowed us to garner a growing recognition in the field of e-business in India, USA, Europe and Australia.

Our core specialization areas are PHP, MySQL and Linux for web application development; and Flash for presentations. We also work on XML.

MagNet aims to provide the right strategies and value added services to help our clients accentuate their presence on-line. We cater to their need to succeed in the networked economy. We do this by closely examining the existing systems and introducing solutions that make it more profitable. At MagNet, we commit our time and resources to providing the most important solutions - those that meet the client requirements.

We invite you to know more about us at:
<http://www.magnet-i.com/>

Contact Us

Have a comment? Thinking about developing a PHP-GTK application? You can get in touch with us on the following addresses:

Nirav Mehta - nirav@magnet-i.com

Vaishali Master - vaishali@magnet-i.com

Piyush Shah - piyush@magnet-i.com

You are free to distribute this document in its original form. You can also excerpt contents and distribute it after giving proper credits.

This document contains material taken from the PHP-GTK Manual.

The MySQL Query Tool code courtesy - Adam Rambousek

All logos and trademarks are copyrights of their respective owners.

MagNet is a customer oriented, technology driven web application development firm based in India.

Are you as excited about PHP-GTK as we are? Let's do something together! Contact us and let's get started!