

Increase traffic: Program a contest!

By Steve Humphrey

Increase traffic: Program a contest!

Steve Humphrey
webmaster@merrymonk.com

Millenniu Market-Place <http://www.merrymonk.com>

This article is Copyright (c)2001 by Steve Humphrey. It may be reproduced verbatim by any means available to anyone who wishes to publish it in any medium - provided 1) NO changes are made without express permission from the author and 2) The author's resource box is ALWAYS included. 3) The author is notified of your intent to publish. A courtesy copy of the ezine, etc. would be appreciated.

Visitors come to your site and don't buy on their first visit. It often takes 4-5 visits before they buy. So, how do you get their business if they don't come back? For that matter, how do you get them to come back?

One answer is to run a contest. Not just any old contest will do! It must be crafted to keep them returning again and again. Lots of smart people will tell you that running a contest will help but they don't tell you how to do it. Here's a blueprint you can alter to suit your purposes.

One of my clients came up with the basic idea and we developed it together as a team. People would get one "entry" into the contest when they signed up for one or more of his newsletters. They would get another "entry" for each person they referred.

It was hugely successful. During the 60 days of the contest, over 45,000 new people had subscribed to my client's newsletters. They were coming back to his site over and over again. Orders for ezine ads poured in.

At the end of the contest period a week-long cruise vacation to Montego Bay, Jamaica was awarded to the winner. Generally, if the prize is valued at over \$500, some authorities in your state may expect to be notified. There are various requirements; check with the appropriate state's agency for details.

We posted a complete set of rules for the contest on the Web site. We announced the contest itself as well as the current front-runners in each issue of my client's newsletters.

Each person entering the contest was assigned a Personal Identification Number (PIN). The PIN was used to connect them with their referrals. Next, a person was given the chance to tell their

friends about the contest.

We made everything as easy as possible for the contestants. All they had to do was supply the email address and name and the scripts did the rest.

They'd be shown the message that was about to be sent to their friend. We were making every effort to be honest with them and not cause any ill will.

When their friend(s) entered the contest, using a PIN we gave them, their "referral count" would be increased. To win, a contestant had to get the most referrals. In the event of a multi-person tie for first place, the software would automatically conduct a random drawing.

During the contest, each contestant was able to use a special Web page to monitor the contest. They'd see the top-ranked contestants and the number of referrals each one had made. They could log in with their PIN and see their own referrals, too.

They were able to come back to the site at any time and enter new referrals. We added a small form that would look up their PIN in case they'd forgotten it. They could send their own emails to friends to tell them about the contest. They could place ezines ads giving contest details and a PIN to use.

Keeping up with all those contestants, who they referred, how many they referred, etc., was a job for a database engine. Since we had MySQL on a server we controlled, we used that. The scripting was done in Perl.

Two Perl modules were needed: DBI.pm and the MySQL module for Perl. The modules are chunks of Perl code that let a script send commands in SQL (the generic language of databases) from the script to the MySQL database engine (program).

You can use a different database; you just need the the database-specific module so Perl can "talk" to it through the DBI module. DBI is needed regardless of which database you use. Specific modules exist for Oracle, PostGres, etc.

We verified the email addresses of all the winner's referrals; we would have to disqualify a contestant if any of the addresses were faked - and award the prize to someone else. We designed the code so that any attempts to cheat actually decreased your odds of winning - and it was all automatic!

Thousands of names and email addresses are in the database. The referral counts have been reset to zero and we are ready to launch another contest at any time we choose.

The programming effort, for the most part is all done. We'll just have to edit the contest page(s) and the rules page to reflect the new prize and the new "start" and "end" dates. We can easily get *at least* as large a subscriber boost with our next contest!

Here's the bottom line: You'll get more visitors, and they'll visit again and again. You get to show them your offer enough times to win their business. No matter what product or service you're selling, an automated referral contest is something you should seriously consider.

=====

Steve Humphrey is an 8-year Internet veteran who is widely regarded as a specialist in HTML, CGI, Perl, SQL, PHP and other programming. His new e-book

"Learn to use CGI in Two Hours: Automate your Web site Tonight" is available across the Internet and from his own Web site: <http://www.merrymonk.com>

[Get-Articles.com](#) : 1000's of reprintable business and internet marketing-related articles.

[Submit your article for reprint.](#)