An Honors College Thesis
Robert K. Price
May, 2001

Robert K. Price, 2001

Dr. Tom Harris, Associate Professor of Management
Advisor
Introduction and Foreword

I was hired by The Center for Mental Health, Inc. in early March of 2000 as an assistant systems’ analyst with the understanding that one of my responsibilities would be to develop the corporate website for use as a marketing tool. This seemed like the perfect opportunity to combine my work experience, my interest in web technology, and my academic pursuits by developing and documenting the site as my Honors Thesis project. I contacted Tom Harris of Ball State University’s Department of Management to ask him to act as my advisor, to which he graciously agreed.

The scope of the project expanded as I found it necessary to familiarize myself not only with the programming and design skills required to create a corporate website, but with other related technologies as well, including an unfamiliar and previously unconfigured web server, a new Microsoft SQL database server, DNS/BIND protocols, name servers, and registering domain names with InterNIC. Overall, I found this an extremely challenging, rewarding and edifying experience. I was able to hone previously attained web design, web development, and graphic design abilities and add several very useful and marketable skills to my repertoire.

Included in this thesis presentation is a compact disc containing all source code, images, and documentation associated with the site, as well as some additional software that would be useful for the development of a similar site. Some of the code may be slightly modified (removal of server names, usernames, and/or passwords) to ensure site security and integrity. It should be noted that the files on the CD will not display correctly if simply opened in a web browser from the CD because they must be run through the PHP script interpreter on a web server to achieve proper display and functionality.

Thank you for taking the time to examine my work. I hope that aspiring web gurus can find some useful ideas, and that laypeople can gain a better understanding of the Internet and web technology.

Sincerely,

Robert K. Price
Ball State University Honors College
Class of 2001
Acknowledgements

I would like to take this opportunity to thank several people for their support, assistance, and inspiration in my creation and completion of this project. So, in no particular order, thank you:

- My parents, David and Jennifer – for being a constant example and inspiration of all that is good about family. I am here and successful because you raised me that way. I love you both!
- Michael Reynolds, President, Spinweb Net Designs – for being my doorway into the web and web stuff and an invaluable technical resource and “free tech support.” You’re all (or at least most of) the geek I want to be.
- Noel Lephart, Systems Analyst, The Center for Mental Health – for taking a chance hiring a know-nothing punk for a techie position and for helping me figure out PHP for NT, IIS, MSSQL. You’re the rest of the geek that I want to be.
- Tom Harris, Associate Professor, Department of Management – for agreeing to be my advisor and sponsor, and for taking the time to discuss IS issues with me during and outside of class.
- Dr. Bonnie McVey, Assistant Professor of Computer Science, St. Norbert’s College – for discovering unexpected programming ability and cultivating an even more surprising enjoyment of programming. I miss you, doc!
- The Staff at The Center – for welcoming me into their organization with open arms (and wagging tongues), and for constantly reminding me that I’m neither as cool or as important as I sometimes like to think I am.
- The Honors College – for allowing its students to choose their own projects and to follow their own interests rather than dictating a traditional thesis paper.

Thank you all, and enjoy!

--R
Table of Contents

About the CMH Site .......................................................... 5

About PHP ........................................................................ 6
  How PHP came into being ........................................... 6
  Why yet another language? ............................................ 6
  Language Syntax ........................................................... 6
  Embedding PHP Code ................................................... 6
  Dynamic vs. Static Web pages ....................................... 7
  Web Application Features ............................................ 8
  CGI/Web server provided variables .............................. 8
  Description .................................................................... 8
  Communication with Databases ................................... 9

Design Elements ............................................................ 9
  Tables ........................................................................... 9
  Common Elements ....................................................... 10

Dynamic Features .......................................................... 10
  Includes/Header/Footer ............................................... 10
  URL Parse/BCT .......................................................... 11
  Jobs Database Integration .......................................... 11
  Counter ......................................................................... 11

Code Examples .............................................................. 12
  Includes/Header/Footer ............................................... 12
  URL Parse/BCT .......................................................... 14
  Database Integration ................................................... 15
  Counter ......................................................................... 16

Appendix A: Glossary/Definition of Terms ...................... 17

Appendix B: Technical Notes/Specs ................................. 18
  Server-Side Hardware/Software Used ......................... 18
  Development Software Tools Used .............................. 18
  Other Hardware/Software Used ................................. 18
About the CMH Site

Before beginning this project, I had some prior experience with web design/development both as an amateur and professionally. I chose to implement the site using a web scripting language known as PHP, which allows for dynamic content as well as easy and convenient editing and updating. It also allows for "sprinkling" and same-file integration with standard HTML tags. These features are especially important at CMH, due to the fact that, aside from myself, there is little technical web knowledge in the company. I wanted to make sure that when I left, my replacement could operate, maintain, and update the site without necessarily having to know PHP. All layout elements that are common to every page are stored in a single file (header.ssi) that is referenced by every other page upon loading. This is a common practice in web design, eliminating the need to make site-wide design changes on every page, an important feature if your site consists of more than four or five pages (consider Microsoft whose sites encompass thousands of pages!).

This site had some unusual and strict content restrictions. As a health care provider, there are client confidentiality policies to which The Center for Mental Health must strictly adhere. There could be no client information or photographs on the site. Also, when describing the group/supervised living homes, no photographs of the locations were permitted, once again to help protect client anonymity. Therefore, all “action” photographs which may appear to contain clients and staff in counseling situations are actually all staff members posing for simulated pictures.

The Center for Mental Health’s CFO, Lori Eliot (sort of the project sponsor), also wanted to use the website as a way to post job openings. Rather than continuously opening, editing, reediting, and updating the jobs page, which could become tedious and confusing, I decided to create a database on the Center’s MSSQL server with which the website could connect to access up-to-date employment listings. I also provided a simple, form-based interface that can be accessed through the corporate intranet (which I also designed and developed).

I received copies of the Center’s promotional literature as well as written descriptions of various programs, service offerings, and facilities, and proceeded to adapt that information to web-friendly content and filtering out repetitive and superfluous information. In April 2000, I began the design and construction of the site in earnest, arriving at a working version around May 15. Before publication to the Internet, the site was initially presented internally to the corporation’s Operating Committee and Executive Board. I solicited feedback and revisions from these bodies as well as individual departmental managers. Near the end of June the site was deployed publicly with the address www.cfinh.org.

Since, the site has undergone almost constant minor revision as services, programs, and locations/facilities have expanded and as managers, coordinators, and supervisors have reevaluated the information needs of their particular areas, but the overall design has remained fairly unchanged.
About PHP

The following explanation is an abridged version of an introduction to PHP technology was written by Stig Sæther Bakken* and explains the technology far better than I could. The full article can be found online at the following address: http://www.zend.com/zendlart/intro.php.

How PHP came into being

PHP started as a quick Perl hack written by Rasmus Lerdorf in late 1994. Over the next two to three years, it evolved into what we today know as PHP/FI 2.0. PHP/FI started to get a lot of users, but things didn't start flying until Zeev Suraski and Andi Gutmans suddenly came along with a new parser in the summer of 1997, leading to PHP 3.0. PHP 3.0 defined the syntax and semantics used in both versions 3 and 4.

Why yet another language?

People often ask "why invent yet another language; don't we have enough of them out there"? It is simply a matter of "the right tool for the right job". Many Web developers found that existing tools and languages were not ideal for the specific task of embedding code in markup. Those developers first collaborated with Rasmus and then later with Zeev and Andi, to develop a server-side scripting language which they felt would be ideal for developing dynamic Web-based sites and applications. PHP was created with these particular needs in mind. Moreover, PHP code was developed for embedment within HTML. In doing so, it was hoped that benefits such as quicker response time, improved security, and transparency to the end user would be achieved. Considering that almost a million and a half sites are currently running PHP (at the time of this article's publication), it would appear that these developers were right.

Language Syntax

Most of PHP's syntax is borrowed from C, although there are elements borrowed from Perl, C++ and Java as well. This article assumes that you are familiar with C's syntax. However, don't panic if you're not.

Embedding PHP Code

To give you an idea of what embedding PHP would entail, consider the following three "hello world" examples, all of which will give the exact same output:

Example 1: HTML alone

Hello, World!

* About the author: Stig's daily work is as a programmer and project manager at Fast Search & Transfer in Trondheim, Norway and Boston, USA.
Example 2: PHP code alone

<? print("Hello, World!"); ?>

Example 3: PHP embedded within HTML

<? print("Hello, "); ?> World!

Web servers supporting PHP will, by default, scan a file in HTML mode. HTML code will be passed over to the browser as usual, up until the server happens upon a PHP line of code. In examples 2 and 3 above, the "<?" tag informs the server that PHP code is to follow. The server then switches over to PHP mode in anticipation of a PHP command. The "?>" tag closes out the PHP mode with the server resuming its scanning in HTML mode once more. Embedding code in this manner is, conceptually, a more fluid approach to designing a Web page because you are working within the output setting, namely an HTML page. Traditionally, you had to fragment the output (i.e. the header, body, footer etc..) and then put it into the code. Now we are inserting the code directly into the output.

From our lone example, however, one might come to ask, "So, what's the difference?" or "Why add extra code when HTML alone would do the trick?"

Dynamic vs. Static Web pages

The "Hello, World" example we chose would certainly not require you to use PHP. That's because it is static, meaning its display will always remain the same. But what if you wanted to greet the world in any number of ways? Say, for example, "Bonjour, World!", or "Yo, World!" and so on.

Since HTML tags are purely descriptive they cannot function as a variable. Nor can they convey even the simplest of uncertainty such as a "Bonjour" or a "Yo". You need a command language to handle variability in a Web page. Based on either a conditional statement or direct user input, a command language can generate the "static" HTML necessary to correctly display a Web page's content.

Let us reconsider example #3. This time we want to let the user decide how to greet the world:

Example 4: PHP embedded within HTML revisited!

<? print("$greeting "); ?> World!

From the above example, $greeting is assigned a value, and together with the comma and the word "World!", this value is sent to the browser.

Dynamic Web page design, however, is more than just about inserting variables. What if you wanted not only to greet the world in French, but also to present the page using the colors of the French flag?
Both a Web page's structure as well as its content can be customized. This means dynamic Web page programming can also entail on-demand Web page building.

**Web Application Features**

One of PHP's oldest features is the ability to make HTML form and cookie data available directly to the programmer. By default, any form entry creates a global PHP variable of the same name.

In the following example, a user name is retrieved and assigned to a variable. The name is then printed by the sub-routine "submit.php":

**Example 5: HTML forms**

```html
<form method="GET" action="submit.php">
What's your name? <input name="myname" size="3">
</form>

submit.php

```

```php
<? print("Hello, $myname!"); ?>
```

**CGI/Web server provided variables**

The variables listed in table 1 are derived from CGI protocols.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$DOCUMENT_ROOT</code></td>
<td>Your Web server's base directory with user-visible files.</td>
</tr>
<tr>
<td><code>$REQUEST_METHOD</code></td>
<td>The HTTP method used to access this page, for example GET or POST.</td>
</tr>
<tr>
<td><code>$REQUEST_URI</code></td>
<td>Full local part of the request URL, including parameters.</td>
</tr>
<tr>
<td><code>$HTTP_GET_VARS</code></td>
<td>An associative array with the GET parameters passed to PHP, if any.</td>
</tr>
<tr>
<td><code>$HTTP_POST_VARS</code></td>
<td>An associative array with the POST parameters passed to PHP, if any.</td>
</tr>
<tr>
<td><code>$HTTP_COOKIE_VARS</code></td>
<td>An associative array with the cookies passed by the browser, if any.</td>
</tr>
<tr>
<td><code>$SCRIPT_FILENAME</code></td>
<td>File name of the top-level page being executed.</td>
</tr>
<tr>
<td><code>$SCRIPT_NAME</code></td>
<td>Local URI part of the page being executed.</td>
</tr>
<tr>
<td><code>$SERVER_ADMIN</code></td>
<td>Server administrator's email address.</td>
</tr>
<tr>
<td><code>$SERVER_NAME</code></td>
<td>Domain name for the server.</td>
</tr>
<tr>
<td><code>$SERVER_PORT</code></td>
<td>TCP port number the server runs on.</td>
</tr>
<tr>
<td><code>$SERVER_PROTOCOL</code></td>
<td>Protocol used to access the page, for example &quot;HTTP/1.1&quot;.</td>
</tr>
</tbody>
</table>
Communication with Databases

Unlike other scripting languages for Web page development, PHP is open-source, cross-platform, and offers excellent connectivity to most of today's common databases including Oracle, Sybase, MySQL, ODBC (and others). PHP also offers integration with various external libraries which enable the developer to do anything from generating PDF documents to parsing XML.

Design Elements

The site was designed using The Center for Mental Health’s corporate color scheme of green and beige. These colors were utilized throughout the site, from the opening graphic to the corporate banner, to the navigation buttons along the left-hand side of each page.

Tables

The page in general uses the HTML concept of tables extensively. A table is a layout tool that enables designers to control the location of content on a page by specifying rows and columns within a table. A designer may stipulate the dimensions of each cell, its background color or image, and even create one cell to "span" other cells.

The opening graphic, which contains a photograph of CMH’s main service building was created as a single image, then, using Macromedia Fireworks, was broken into several images and arranged within a table. This allows for faster loading (several small images rather than one large one), as well as the ability to assign links to different "locations" within the graphic. Below on the left is the opening graphic of the site as it is seen on loading. On the right is the same graphic, but with red lines indicating the cells of the table and therefore, individual images.
Tables were also used for the standard layout of the web pages. Each page on the site was actually one large table consisting of one row and two columns. This allowed the navigation elements on the left side to be displayed consistently and correctly regardless of the viewer's monitor resolution (800x600 vs. 1024x768). This practice of tabling also allows for easier site maintenance and greater layout flexibility.

Common Elements

Another important design element of the site is the common elements displayed on each page. The most obvious of which is the navigation button group, located along the left side of each page, which is included to help facilitate site navigation (Fig. 3). Each button is a separate graphic, and is set to link to the appropriate area, i.e. the services button hyperlinks to /services/. Both text and icons were included for each graphic to increase ease of navigation. In addition, the CMH banner is displayed on every page, as is the copyright notice and webmaster mailto: link.

Also included on each page is the Yahoo! style navigation “bread crumb trail.” This tool, created by parsing the URL and creating links to parent levels in the directory structure. In the example below, blue text indicates a link to a parent directory, i.e. clicking on services will take the browser to http://www.cfmh.org/services/. This tool, in one form or another, is fairly common on the Internet, and is an extremely useful navigation aid.

Example: 6 Bread crumb trail for http://www.cfmh.org/facilities/CSS/GRIP/index.php3
home >> facilities >> CSS >> GRIP

Dynamic Features

Includes/Header/Footer

The common layout of each page is stored in a single file and is included in every page upon loading. This practice is known as using includes, or server-side includes. The file header.ssi (.ssi stands for server-side include) contains the HTML code to create the green stripe and buttons on the left side of the page as well as the banner and the bread crumb trail. Closing “cleanup” HTML is contained in the footer.ssi file.
URL Parse/BCT

Another dynamic feature is the URL PARSE/Bread Crumb Trail located on each page. Upon load, the URL of the page is broken down into its component directory structure and displayed on the page. The advantage of implementing this dynamically is that a single file (url_parse.php—demonstrated later) can be referenced by each file (or in the document header) rather than hard-coding the links into every individual page.

Jobs Database Integration

The website interfaces with an MSSQL database which contains job listings for currently-available positions. Each time the page is loaded the database is accessed, retrieving all current job listings, including title, date available, description, requirements, hours per week, and requirements, formatting the list and outputting the information to the screen. Figure 4 below shows the page displaying the jobs, while Figure 5 shows the add job and maintenance forms on the corporate intranet.

Counter

Jerry Landers, Director of Expansion, requested a way to track the number of visitors (other than CMH employees and test loads) that had accessed the site. He was not concerned with having the hit count is displayed on the site, and I felt that it would detract from the attractiveness. I developed a simple counter that excluded interior web traffic and simply incremented a counter in a text file.
Code Examples

This section contains HTML and PHP code examples taken directly from the CMH website. The text in the examples is color-coded consistent with the display scheme used by Editplus (the text editor used to create the majority of the site). HTML tags and native statements are shown in blue. PHP variables are shown in teal. Built-in functions are shown in red. Assigned values and information passed to functions are shown in pink. Comments (non-executed code) are shown in green. Comments may be denoted by // Commented Code, # Commented Code, or /* Commented Code */.

Includes/Header/Footer

Below is the HTML code contained within the header.ssi file. All common layout features are contained within this file and footer.ssi, as well as the commented copyright notice which can be read if a user selects View → Source (View → Page Source on Netscape Navigator) from his/her internet browser.

Server-side includes are referenced with the following PHP command:

```php
<? include("$SSI_DIR/header.ssi"); ?>
```

where $SSI_DIR is a variable containing the full local path to the directory where the SSI files are located. Note that all other paths (such as those referencing images) are relative to the “web root” directory (/). An included SSI file may contain other include statements, as shown in header.ssi below:

```html
<!--header.ssi-->
<!-- ################################################################################################################### -->
<!-- # All design and layout, as well as original graphics/artwork and photography by Robert K. Price # -->
<!-- # All content on the Center for Mental Health web site and server including the elements of design and layout unless otherwise noted is copyrighted # -->
<!-- # material and protected by trade dress and other laws. # -->
<!-- # By viewing the source of this document, you signify you have read the above # -->
<!-- # copyright notice and accept the terms of use. # -->
<!-- # Created: May 20, 2000 – April, 2001 header ssi # -->
<!-- ################################################################################################################### -->

<?

#####################################################################################################################
## INITIALIZATION OF CONSTANT PATHS ##
#####################################################################################################################

$SSI_DIR = "C:\website\SSI"; # Path to server-side includes directory

?>
<!--Style Sheet-->
<STYLE>
//--
A:link {text-decoration: none; color: blue}
A:active {text-decoration: none; color: red}
A:visited {text-decoration none; color: blue}
#bod)1ext
 Geneva, Verdana, sans-serif; font-size: 10pt; text-align: justify 

</STYLE>

<BODY LEFTMARGIN="0" TOPMARGIN="0" MARGINWIDTH="0" MARGINHEIGHT="0">

<!--This table is used to align navigation content within the color strip down the side of the page-->
<!--The closing HTML tags are found in the footer.ssi file-->
<TABLE BORDER=0 VALIGN=TOP ALIGN=LEFT>
<tr>
<TD WIDTH="150" BGCOLOR=#336633 VALIGN=TOP ALIGN=LEFT>
<IMG SRC="/images/cellbackground.jpg" WIDTH=150 HEIGHT=20>
<br>
</TD>
</tr>
</TABLE>

<!--Navigation Bar-->
<CENTER>
<!--CFMH Banner-->
<DIV ALIGN=CENTER><IMG SRC="/images/banner2.gif" ALT="The Center for Mental Health" WIDTH="425" HEIGHT="90"></CENTER>
</DIV>
<? include("SSI_DIR/url_parse.php"); ?>

<!--BEGIN BODY CONTENT--ALL TEXT ABOVE CONTAINED IN header.ssi-->
<!--BEGIN BODY CONTENT--ALL TEXT ABOVE CONTAINED IN header.ssi-->

13
URL Parse/BCT

The following code receives the requested URL/URI and parses it to create the bread crumb trail.

```php
<?
// url_parse.php
// If it's root directory print directory tree
if($PATH_INFO != "/" && $PATH_INFO != "index.php")
{
    $tmp = explode("/", $PATH_INFO);
    // If the URI ends with / pop off the last empty element of the array
    if($tmp[count($tmp)-1] == "index.php")
    {
        for($i=0;$i<count($tmp)-1;$i++)
        {
            $newarray[] = $tmp[$i];
        }
        $tmp = $newarray;
    }
    $numlevels = count($tmp);
    for($i=0; $i<$numlevels; $i++)
    {
        $tmp = $tmp[$i];
        // All URIs start with a / so if the first element of the array is empty, make it home
        if($tmp == "")
        {
            $tmp = "home";
            $build_url = "/";
        }
        else
        {
            $build_url .= $tmp."/";
        }
        // If the last level is a document strip mime type (file extension)
        if(substr($tmp, -4) == ".php" || strpos($tmp, ".")
        {
            $tmp = substr($tmp, 0, strpos($tmp, "."));
            // replace all underscores with spaces
            $tmp = ereg_replace("_", " ", $tmp);
        }
        // If last level, don't make a link
        if($i == $numlevels - 1)
        {
            $bread_crum_trail .= "<B> $tmp </B> ";
        }
        else
        {
            $bread_crum_trail .= "<A HREF="$build_url">$tmp</a> &gt;&gt; ";
        }
    }
}
else
{
    $bread_crum_trail .= "<b>home</b>";
    $tmp = "Welcome to CMH";
}
print("<FONT SIZE="-1">$bread_crum_trail</FONT><BR><BR> ");
?>
```
Database Integration

As described earlier, the site uses a database tie-in to track the job listings for the company. The values below were changed from the values that were used for security reasons. The following code is included at the beginning of the /jobs/index.php3 file and provides the database connection:

```php
<?

// jobs_db_connect.php
// Set Up Database Values
$Server = "SERVER NAME";
$Username = "DEFAULT WEB USER";
$Password = "PASSWORD";
$Database = "DATABASE";
$table = "TABLE";

// Connect to Server
mssql_connect($Server,$Username,$Password) or die("Error: Cannot Connect to Server");

// Select Jobs Database
mssql_select_db($Database) or die("Error: Cannot Connect to Database");
?>

After the connection is established, the database is accessed using SQL queries such as:

```php
// Grab All Info From 'TABLE' Table
$Query = "SELECT * FROM $Table";
$Result = mssql_query($Query);
```

The result of the query (an array called $Result) is then parsed and manipulated into useful form for display.

```php
while($Row = mssql_fetch_row($Result)) // Loops through each result
{
    // Get Columns
    $Job_Title = $Row[0];
    $Job_Date = $Row[1];
    $Job_Desc = $Row[2];
    $Job_Req = $Row[3];
    $Job_Hrs = $Row[4];
    $Job_F = $Row[5];

    // Remove time stamp from $Job_Date since it's not used
    $Job_Date = ereg_replace("12:00AM", ",", $Job_Date);

    Once all database operations have been completed, the database connection must be closed:

    // Close Database Connection
    mssql_close();
```
Counter

The following code executes the counter as described in the "dynamic features" section. Far from complex, this script merely opens a file that already contains a number (counter.txt), reads that number, increments it, then writes it back to the same file.

The only complex feature of this script is the section to test whether or not the computer requesting the page is from CMH. To exclude internal viewers from the hit count, the ereg command, which stands for e-regular expression, was used. This command retrieves the users IP address from the http headers (passed by all browsers) and determines whether or not it begins with '192.168' which would indicate that it comes from a computer within CMH’s network. If it does, the script does nothing, otherwise, the counter is incremented ($count++).

```php
<?
// counter.php
$counterFile = "E:\Apps\IETN\WWWroot\website\SS\counter.txt"; // Location of the textfile

if(ereg("^192.168.*", $REMOTE_ADDR)) // Environment variable containing IP address
{} // Do nothing, it is a local IP (Internal request)
else // Get previous hit count
{
    $counterFile = fopen($counterFile, "r"); // Create file handle for reading
    $counter = fgets($counterFile, 255); // Read in old value
    fclose($counterFile); // Close file handle

    // Increment count
    $count++;

    // Output new count to file
    $counterFile = fopen($counterFile, "w"); // Create file handle for writing
    fputs($counterFile, $counter); // Put new value back into file
    fclose($counterFile); // Close file handle
}
?>
```

*The ^ mark tells the function that this pattern must be at the beginning of the string. The .* mark is a wildcard which means "anything." Therefore, 192.168.1.10 and 192.168.7.10 would not be included in the count, but 147.226.192.168 would be.*
Appendix A: Glossary/Definition of Terms

DNS (Domain Name System) the way that Internet domain names are located and translated into Internet Protocol addresses. A domain name is a meaningful and easy-to-remember "handle" for an Internet address.

HTML (Hypertext Markup Language) the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page.

HTTP (Hypertext Transfer Protocol) the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

PHP (Personal Homepage Tools) – a script language and interpreter that is freely available and used primarily on Linux Web servers.

SQL (Structured Query Language) a standard interactive and programming language for getting information from and updating a database.

URL (Uniform Resource Locator) is the address of a file (resource) accessible on the Internet.

WWW (World Wide Web) all the resources and users on the Internet that are using the Hypertext Transfer Protocol.

mailto a link set up within a web page to automatically open the web browser's built-in email client to compose an email to the specified address.

**All definitions (with the exception of mailto) quoted from www.whatis.com**
Appendix B: Technical Notes/Specs

Server-Side Hardware/Software Used

<table>
<thead>
<tr>
<th>Category</th>
<th>Resource</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Compaq Proliant 1600</td>
<td><a href="http://www.compaq.com">www.compaq.com</a></td>
</tr>
<tr>
<td>Server Specs:</td>
<td>Dual PII 500Mhz Processors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1,024 MB RAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAID5 Array (5 x 9.8GB SCSI2 HDDs)</td>
<td></td>
</tr>
<tr>
<td>Connection:</td>
<td>Dedicated IQuest T1 Line</td>
<td></td>
</tr>
<tr>
<td>Operating System:</td>
<td>Microsoft BackOffice 4.0 SP 6</td>
<td><a href="http://www.microsoft.com">www.microsoft.com</a></td>
</tr>
<tr>
<td>Web Server:</td>
<td>Microsoft Internet Information Server 4</td>
<td></td>
</tr>
<tr>
<td>Database Server:</td>
<td>Microsoft SQL Server 7</td>
<td></td>
</tr>
<tr>
<td>DNS Server:</td>
<td>Microsoft DNS Manager 4</td>
<td></td>
</tr>
<tr>
<td>PHP Interpreter:</td>
<td>PHP 3</td>
<td><a href="http://www.php.net">www.php.net</a></td>
</tr>
</tbody>
</table>

Development Software Tools Used

<table>
<thead>
<tr>
<th></th>
<th>Resource</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Editor:</td>
<td>EditPlus 2.01a</td>
<td><a href="http://www.tucows.com">www.tucows.com</a></td>
</tr>
<tr>
<td>Image Editor:</td>
<td>Photoshop 5.5</td>
<td><a href="http://www.adobe.com">www.adobe.com</a></td>
</tr>
<tr>
<td>Other:</td>
<td>Macromedia Fireworks 3.0</td>
<td><a href="http://www.macromedia.com">www.macromedia.com</a></td>
</tr>
</tbody>
</table>

Other Hardware/Software Used

<table>
<thead>
<tr>
<th></th>
<th>Resource</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Camera:</td>
<td>Kodak DC280 Zoom</td>
<td><a href="http://www.kodak.com">www.kodak.com</a></td>
</tr>
</tbody>
</table>