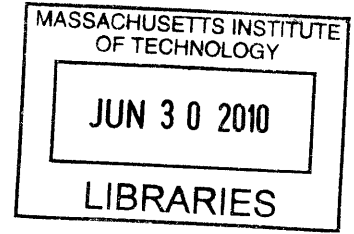


Use of Modular Web Developments to Ease the Language Barrier of
Hypertext Processing and Structured Query Language

By

Jarrell Dwayne Johnson



SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

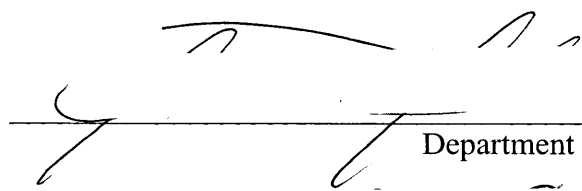
JUNE 2010

ARCHIVES

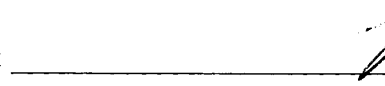
©2010 Jarrell D. Johnson. All rights reserved.

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part
in any medium now known or hereafter created.


Signature of Author: _____


Department of Mechanical Engineering
May 10, 2010

Certified by: _____


David R. Wallace
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: _____


John H. Lienhard V
Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

Use of Modular Web Developments to Ease the Language Barrier of Hypertext Processing and Structured Query Language

By

Jarrell Dwayne Johnson

Submitted to the Department of Mechanical Engineering
on May 10, 2010 in partial fulfillment of the
Requirements for the Degree of Bachelors of Science in
Mechanical Engineering.

ABSTRACT

Four modules were developed with the extensive use of Personal Home Page (PHP) Hypertext Processing and Structured Query Language (SQL) in order to improve web-integrated capabilities of events planning, user account management, mass electronic mailing systems, and form and survey creation. Prerequisites to using the modules are reduced to only needing a pre-configured database and its associated credentials.

In using a multi-step process to install each module, the user is able to specify minimal details that configure the functions of the web application.

By developing a user accounts management system and integrating the other three web applications as sub-modules, each system introduces administrative control panels with user-controlled access capabilities. The results of the web modules show that both self-hosted and remote websites are capable of integrating the systems and that users with either no experience or some minimal experience are capable of using the modules.

The modules can be obtained from the following link:

<http://www.jarrelljohnson.com/modules>

The page provides a brief overview of each module and instructions on how to install the module. The pages also provides a direct download link to for each module. For further instructions, please reference the individual instructions manual within the module.

Thesis Supervisor: David R. Wallace
Title: Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

First, I would like to thank several key individuals--Prof. David Wallace, Prof. Alexander Slocum, and Lori Hyke--that have helped to open the door for me to explore a senior thesis that has enriched my knowledge of web design, PHP and SQL programming, and the use of programming design software. Without the support and the limitless freedom provided to explore this topic, I would not have been able to further my understanding of the design material needed to produce integrated web-based products.

I would also like to thank one key individual who has helped guide me for four years at MIT, Prof. James H. Williams. Without him, I would have been clueless as to what projects I would have liked to pursue, what classes I should have taken as electives to help diversify my education, or what I was going to do after my undergraduate career. In the short amount of time that I had to interact with him, Prof. Williams has given me ample support and helped me to gain the confidence I need to excel beyond my dreams.

Lastly, I would like to dedicate my senior thesis to my mother, Angela Ware. For the first time our family will see someone graduate from college, and I am happy that I am able to provide the experiences of attending MIT as her reward for the many years of encouragement and un-ending dedication to believing in me. Her belief in me has helped me go farther than I have imagined and I am greatly thankful for that.

TABLE OF CONTENTS

INTRODUCTION	5
<i>Potential Usage.....</i>	<i>6</i>
<i>East of Maintenance.....</i>	<i>7</i>
<i>Integration Capability.....</i>	<i>8</i>
<i>Capability of Module Developments.....</i>	<i>9</i>
MODULAR DEVELOPMENTAL PROCEDURES.....	10
MODULE INSTALLATION.....	11
MODULAR FUNCTIONS AND CAPABILITIES.....	15
<i>User Accounts Management System.....</i>	<i>15</i>
<i>Mass Electronic Mailing System.....</i>	<i>20</i>
<i>Calendar and Events Management System.....</i>	<i>22</i>
<i>Form and Survey Creation System.....</i>	<i>26</i>
RESULTS OF ONLINE TESTING	29
CONCLUSION	31
APPENDIX A - SCRIPTS FILE TREE.....	32
APPENDIX B - INSTALLATION SCRIPT EXCERPTS.....	36
APPENDIX C - MISCELLANEOUS CONTROL PANEL SCRIPTS	57

INTRODUCTION

Currently, the Internet serves as a pillar for communications, personal displays, multimedia, and even advancements in knowledge. As a result, the desire to make things easier for an individual tends to lead to introducing an online component. This includes things like shared documents and other information, web-based surveys and forms, and electronically delivered notifications. Such dependence on the online media provides a unique opportunity for individuals who are comprehensive in web script writing.

There are many ways that an individual can learn how to write scripts to produce online media: open-source websites such as Wikipedia introduce search capabilities for each web language; other sites such as w3schools provide step-by-step walkthroughs that help guide a user through the learning process; and alternatively, anyone is capable of registering for classes at most local Colleges and Universities that teach the language and its relation to other languages. Regardless of whatever the process used to learn the language is, the problem is still apparent that much time is required in order to gain enough knowledge to apply the language.

Beyond gaining knowledge and understanding of a language, there is one major challenge with web programming: most effective web pages require the comprehensive knowledge of a least two independent languages in addition to the standard HTML5 base. As such, many individuals who understand how to make a simple web page are not capable of producing more dynamic web products.

The development of web modules stems from the same mentality that goes into the design of an engineering product. For example, when an individual found it difficult to carry materials that were too heavy a product was designed and developed to ease those difficulties. The same applies with web design: when someone wants to collect data about a particular event,

they develop an interface that will allow them to collect said data. In essence, they must come up with some mechanism that will assist them in achieving the end goal. Such developments are capable for either one of three reasons or a combination of 1) being knowledgeable in multiple programming languages, 2) having programs that facilitate web design, or 3) working as a team of individuals with each person specialized in a particular language.

When determining how to approach the development of web modules, the key questions asked were how can you best assist individuals in achieving their own personal web-based goals without compromising the personal touch. And in producing the modules, how do you determine what modules are most important. Additionally, how can web modules be developed such that they can be integrated into the use of any website.

For deciding what applications of web programming are relevant, it was important to stay away from designing modules that are built on ease of scripting, script familiarity, or low usage. For that reason, the basis for the decisions was ease of maintenance, potential usage, and integration ability.

POTENTIAL USAGE

One issue that was to be addressed in the development of web modules is the expected usage of the module. When looking at the most frequent uses of the Internet, communications serves as the primary use. This is often in the form of forums, blogs, and most notably electronic mailing. To capitalize on this, the need for visitor tracking systems and mass mail forwarding appear to be key developmental components. Such components enable the web hosts to communicate with each user registered for their site, both through site visits and by electronic

mailing systems. In addition to integrating the system, the modules are capable of tracking visitation usage statistics and other visitor data.

In researching other widely used web tools, calendar and events management appears to be another key tool not only for personal management systems but also for student groups, corporate groups, and small businesses. The development of a calendar management system would allow for a user to maintain a list of their events and the data associated with each event long after the event had occurred. As a part of that, the information can be easily incorporated into any page in the form of a calendar or a list with the use of predetermined code.

The last module that was identified was one that serves as one of the most needed resources: a mechanism for the creation and personalization of forms and surveys. Most student groups, corporations, small business, etc., need a way of collecting information. Whether it be the collection of contact information, new applicants, or feedback surveys, a mechanism that eases the interface between PHP, SQL, and the online form is one of the most essential modules that can be developed.

As such, four modules were determined: user accounts management, calendar and events management, mass electronic mailing systems, and forms and survey creation.

EAST OF MAINTENANCE

Maintenance is a key issue with implementing web design. It is one of the most important factors to be considered when writing script. For this reason, each web module was to be designed with 1) the intention of using codes that are not currently being depreciated in the development of newer versions of web programming languages and 2) the ability for a user to reconfigure the module through the installation process.

To account for depreciation, each module was to be designed using a minimum of five languages: Structured Query Language, Personal Home Page Hypertext Processing, Cascading Style Sheets, Javascript, and the standard HyperText Markup Language. In addition to the importance of avoiding depreciation, the use of five web programming languages assists in ensuring browser compatibility.

By using all five languages, information is capable of being stored, dynamically altered, and shared between multiple files. This usage also doubles as a process that improves web security because either the page cannot be read by a single-language processor or it limits the browser with the implementation of page headers.

While the maintenance of the web module is the biggest concern, it is the one key issue that can be addressed within any module. In deciding what modules were to be developed, this key factor was alleviated in an effort of using it to confirm whether the module could be created rather than whether or not it should.

INTEGRATION CAPABILITY

The integration of each module was to serve as the most challenging part of the project: developing web modules that enable script creation while also seamlessly integrating into a pre-existing site. The known limitation that would stem from the development of each module is that each host site requires access to the PHP programming language. Beside the PHP requirement, it was important to not introduce other limits to integration.

Over the course of writing the script for each module, it was to be tested in the four most used browsers, Internet Explorer, Firefox, Google Chrome, and Safari. In addition, the modules were tested cross-platform, on both Mac OSX and Microsoft Windows 7. Ensuring that each

module worked between all platforms was important, not only for the various users who can install the module and administer it, but to ensure usability for any user who visits a web site with the module installed.

CAPABILITY OF MODULE DEVELOPMENTS

Each module was built with specific capabilities to assist, not only in ensuring the above criteria, but also to provide an effective means of increasing usage per web site. The four modules were built with features that allowed the following:

The user accounts management system allows an administrator to introduce individual user accounts that are created on an as-is basis. Incorporated into the accounts system is a user groups system that can then be coupled with a self-generated header to limit access to a particular site based on the defined group. In addition to limited access or redirection, the system stored the data into PHP session variables, which can be referenced on any page of the site until the user logs out.

The mass electronic mailing system allows an administrator to send personalized messages to users registered for the site. The messages can be sent to specific user groups defined in the user accounts system or to every user registered. The type of message can also be customized in either rich text format or HTML scripting.

The calendar and events management system allows the administrators to create an event profile and post it to a calendar that can be integrated into any PHP script page. The events are automatically synced with the calendar and introduces the capability to provide user interactions with events through an assistive administrative interface.

The forms creation service allows an administrator to produce easy to use forms that are integrated with a personalized table in a SQL database in accordance with the user preferences. The form is highly customizable from required or readonly questions to multiple textbox and input types. Each easy-to-make form produces its own script for the integrated form, processing scripts, and table creation. The form can then be simply activated by copying the link into the necessary page.

These capabilities combined can greatly increase web functionality and together can be combined to create a massive management system. All that is required of the administrator is the knowledge of their databases in order to install the modules.

MODULAR DEVELOPMENTAL PROCEDURES

The modules within this project were developed in series, rather than in parallel. Because each module will implement similar functions and user configurations, it was important to construct the modules in such a way that ensures the techniques work prior to implementing the structure in several other versions. In addition to that method of implementation, it was important to limit the administrative capabilities within each module to the user that installs the application (and any other users that are designated by the installer). Of the four, this put one module as the key to the entire system: user accounts management.

The user accounts management module serves as the primary module and is required before an individual can install any other module. While this may appear to breach the goal of integration ability, it serves as a facilitator that eases integration and improves security. Additionally, it reduces the input required of a user and introduces more advanced monitoring systems. This is discussed in more detail in the respective sections.

To produce the modules, a very basic text editor was used to write the script. Using a text editor as opposed to an established program enabled a better monitoring system for the integration of the five languages used to write each individual module. It also helps to ease the preferential treatment that is introduced by using applications built to write a specific language.

The modules were produced in the following order: user accounts management, mass electronic mailing, calendar and events management, and forms and survey creation. At the face of each modules installation and functional execution there are many similarities, however, each module varies greatly in its background coding. The similarities are based on the required information used to access the SQL Databases and to provide a similar experience for each module. Over all, the modules come together in a total of one-hundred and forty-four files (see “Script File Tree” – Appendix A). In the subsequent sections, you will be introduced to the installation procedures within each module, the functions they are capable of, some of the scripts that are used to implement the functions, and the review of its testing. To reduce the display of redundant coding, snippets of coding will be shown in order to show the integration of coding and commentary and to explain particular key scripts of the module.

MODULE INSTALLATION

The proper installation is an integral part of the implementation of each module. To facilitate the installation, the script was written in order to provide the most simplistic methods. For each installation, no more than twelve questions are asked of the installer—all of which should be previously provided by the hosting services used to host the module.

Upon installation, all modules, with the exception of mass electronic mailing, will require the name of the SQL database, the address to the SQL server, the username and password used to

access the database, the directory where the module is installed, and basic information such as the installer's time zone and e-mail address. This information, in addition to the module's personalized information is scripted into a PHP file that produces the configuration files necessary in order to process the module.

In addition to the above-mentioned information, user accounts management system asks for a redirection link, whether the user requests accounts to be verified via email, the properties for the individual user passwords, and the number of user groups wanted (Figure 1). Together these inputs complete the needed installation information.

Please read the instructions prior to beginning the installation.

Installation Information	General Preferences
Database Name: <input type="text"/>	Website Domain: <input type="text"/>
Database Server: <input type="text"/>	Redirection Link: <input type="text"/>
Database Username: <input type="text"/>	Require Verification?: <input type="text"/>
Database Password: <input type="text"/>	Minimum Password Length: <input type="text"/>
Install Directory: <input type="text"/>	Maximum Password Length: <input type="text"/>
Admin E-Mail: <input type="text"/>	Log Account Logins?: <input type="text"/>
Time Zone: <input type="text"/>	Number of User Groups?: <input type="text"/>

Figure 1. Screen Shot of the Account Management Installation

The redirection link is the default link used to redirect a user in the event that they are required to log in without having visiting a page that required authorization. For example, if the installer visits the administrative control panel they are asked to sign in and will be directed to the control panel upon logging in, however, if the user were to click a link from the homepage to log in then the user would be redirected to the default link entered here.

The verification of the email address is to ensure that the individual signing up is using a valid e-mail address. If the installer would like a more secure login module, then e-mail verification is essential. By choosing to require a user to verify their e-mail, the module ensures that each account is valid through the implementation of two additional steps prior to activating an account: sending an activation link to the e-mail address entered and requiring a second login after registration.

The password properties are used as limits for user registration and the log login and user group options are used to enable additional settings within the administrative control panel. The password properties are only monitored during the installation process and during the registration of new user accounts. Simply editing the configuration file that is returned at the conclusion of the installation can allow changes to these properties. Enabling log login will allow the administrator to view a log of times for User Accounts and IP Addresses associated with each user registered for the site. This function makes it easier for the administrator to ban a user if there is suspicious activity taking place. The user group option allows the administrator to create groups and define access levels for each group. By default, two groups are already created: User and Administrator.

Depending of whether user groups are enabled, either one or two steps remain. If user groups are enabled, the installer is asked to specify the names of the groups. Otherwise, the installer is asked to create the administrative account that will be used to administer the module. Once the information is provided, the script writes the final information to the configuration file, produces a backup text file, and makes a sample header file with instructions on how to make a page password protected (see “Installation Script Excerpts” – Appendix B).

The installation of the calendar and events management system and survey and forms creation systems follow similar installation procedures, however, they introduce one additional step due to the user accounts pre-installation requirement. Prior to beginning the installation process, these installers ask for the web address to the user accounts directory (Figure 2). While it introduces only one additional input from the installer, it incorporates an entirely new script that must import the data from user accounts and stores the information into its own personal configuration file within its own directory.

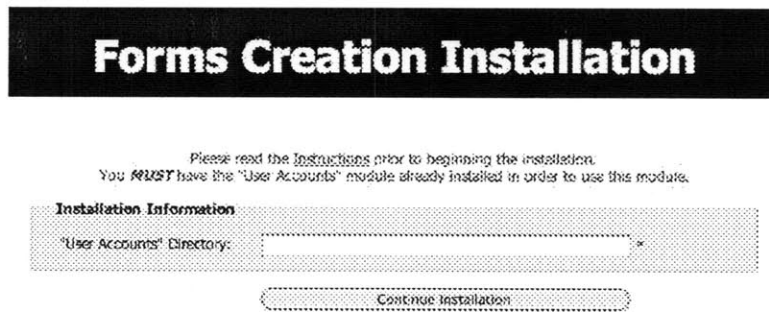


Figure 2. Screen Shot of Forms Creation Installation

After exporting the information, the two installers differ greatly. The forms and survey creation installer does not require any additional information other than the information that is similar among all three installers. The calendar and events management system, on the other hand, introduces two more features: the sponsor and category list. These two features are used to associate each event with a sponsor or an event category if the installer so chooses. The installation of these features work similarly to the user account's user group feature in that the administrator selects the number associated with each feature and is introduced to an additional step where they supply the names. This information is then stored into their own respective tables and is referenced in the administrative events add form which is explained later.

Lastly, there is the mass electronic mailing system. This system varies greatly from the other three module installations. The mass electronic mailing system has only one step needed for installation: the web address to the directory where the user accounts management system is installed. The rest of the code gathers the information from the configuration files and copies the properties into its own personal file.

Prior to installation, there are several files missing that prompt the users with an error. During installation, all of those files are produced and thus the requirements are fulfilled. So for each module, after completing installation it is immediately ready for use.

MODULAR FUNCTIONS AND CAPABILITIES

USER ACCOUNTS MANAGEMENT SYSTEM

Of the four modules, the user accounts management system is the second least complex module. While it is not the most complicated, it is the most important. The user accounts module makes use of fifty individual scripted files in order to be properly executed. The files vary greatly between simple HTML and PHP files. This high number of files is due to the versatility that is required by the redirection mechanisms used within the script in addition to the many status variations between users.

Take, for example, the various stages of an user account: user registration, user verification, user login, and user control options. Each of these stages introduces a different level of a site visitor's interaction. For this reason, the module appears complicated. However, the pages themselves are very simple and highly redundant. While this appears to be an ineffective means of providing redirection and login services, it is the most efficient way to provide multiple interfaces for the same functions.

The greatest complexity, however, comes from the administrative control panel. The control panel allows an administrator to perform many actions on users. Those actions include adding a user, modifying and user, deleting a user, recovering a user, expunging all deleted users, banning a user or IP Address, lifting a ban on a user or IP Address, or changing a user's assigned group.

In adding a user, the administrator is introduced to a similar form that was used for the creation of the administrator account during the installation of the module. Basic information, including user name and password, the user group, and the e-mail address are asked of the admin. It is then at the discretion of the administrator to send an e-mail to the new user, prompt them to change their password, or require them to confirm their e-mail address. Once the user is created, it is then displayed in the master list and included as an option for modification. The most important factor here is the delegation of the user groups. If the administrator creates another administrator, the new user is capable of performing any option within the control panel.

To modify a user, the administrator is prompted to search for the user first. This is to limit the bandwidth in the event that there are hundreds of users in the system. Once acquired, the administrator is prompted with the same form as presented in the add user interface. The difference, however, is that the form is already pre-filled with the users information. Any changes made here execute the same scripts as if the user was being added.

The remaining functions are a bit different as they are all grouped as the miscellaneous options. To ban a user, the administrator has the option of inputting an IP Address or selecting a username. If a username is selected, the username is entered into a banned member table. Similarly, an IP Address can also be entered into the table. The biggest difference between banning the IP Address instead of the username is that IP Address limits the computer from

registering for another account. This action can be undone by unbanning the username of IP Address. Because a ban does not delete a member, the information is still stored in the user account database.

If an administrator wishes to completely remove a member, they may choose to delete the user account from the database. This action will later affect the user by simply stating that the account no longer exists rather than redirecting to a banned user page. Similar to the ban system, a user may also be recovered from the trash so long as they are not expunged prior to recovery.

To test the implementation of password protected logins, the sample header that was generated from installation was placed on several pages of the test website during production.

The installation generated the following code for my host:

```
<?php
// prevent caching of the web page
header("Expires: Sat, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: post-check=0, pre-check=0",false);
session_cache_limiter();
session_start();

// require the configuration and the functions files
require('/home2/jayworld/public_html/thesis/users/users-config.php');

// check to see if the user is logged in
if (!$_SESSION['user_name']) {
    $_SESSION['temp_redirect'] = "http://$domain" . $_SERVER['PHP_SELF'];
    header("Location:http://www.jarrelljohnson.com/thesis/users/acct-login.html");
    exit;
}
?>
```

The “//” represent the commented regions that help the user understand what the code is doing. To begin, the headers for the webpage are sent to the browser. By defining the page date out of range, the computer will delete the cache at the end of the session. This helps alleviate data storage of the login credentials for both the user and the SQL database.

After sending the headers, the session is started. By using sessions, the script introduces the ability to track information from page to page without using multiple forms. The configuration files are then incorporated into the page's code. This allows code that is duplicated on multiple pages to be included without having to lengthen the amount of space in the actual document. Like the header, these files are generated during the installation process. Lastly, the session variables are conditionally tested to verify that a user is logged in prior to viewing the page. If they are not, they are redirected to the login page. If they are, nothing is done and the rest of the page is generated.

To determine what action must be taken for a user, a redirection script is used. The redirection script is the most complex of all user account scripts. The excerpt of the redirection script below shows the order in which redirection links are assigned.

```
// make the connection to the database
$connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
$db = @mysql_select_db($dbname,$connection) or die(mysql_error());
// if activation page is being used, adjust the verification
if ($_POST['activate'] == "YES") {
    $sql ="UPDATE $tblname
        SET verified = 'YES' WHERE username = '$_POST[username]";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
}

// check to see if the user is coming from a page other than the login page
if ($tempdir != "") {
    $sql ="UPDATE $tblname
        SET redirect = '$tempdir' WHERE username = '$_POST[username]";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
} else {
    $sql ="UPDATE $tblname
        SET redirect = '$redirect' WHERE username = '$_POST[username]";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
}

// set the session variables
SessionVariables($sqlserver, $dbuser, $dbpass, $dbname, $tblname, $username, $password);

// check to see if the user is required to change the password
if ($_SESSION['changepass'] == "1") {
    $_SESSION['redirect'] = "$directory/acct-changepass.html";
}
```

```

}

// check to see if the user is required to activate the account
if ($_SESSION['verified'] == "NO") {
    $_SESSION['redirect'] = "$directory/acct-notactivated.html";
}

// check to see if the user is banned
$sql = "SELECT * FROM banned";
$result = @mysql_query($sql,$connection) or die(mysql_error());

while ($sql = mysql_fetch_object($result))
{
    $banned = $sql -> no_access;
    if ($username == $banned || $_SERVER['REMOTE_ADDR'] == $banned) {
        require('acct-banned.html');
        exit;
    }
}

$lastlogin = gmdate("Y-m-d");

// update the last login of the user
$sql = "UPDATE $tblname
        SET last_login = '$lastlogin' WHERE username = '$_SESSION[user_name]'";
$result = @mysql_query($sql,$connection) or die(mysql_error());

if (($_SESSION['redirect'] != "$directory/acct-loginerr.html")
    && ($loglogin == "YES")) {
    require('script-loglogin.php');
}

// redirect the user based on the executed code
header("Location:$_SESSION[redirect]");

```

The above code demonstrates the many conditions that need to be tested when a user is being evaluated. These steps must occur in series such that the most critical redirection link is assigned last. The first step is to test whether or not the user is activating their account. If they are, their account is updated to reflect so. From there, the redirection page is determined. If the user is coming from a previous page, the redirection is set. Otherwise, the redirection link is set to the default link provided during installation. After establishing the redirection, the user is tested for changing their password or non-activated credentials. If either is true, the redirection is

reset to the specified notification page. The next step results in further investigation into whether the user is banned, and if so, the redirection is again reset. From there the information is logged if the administrator requested the log login algorithm during installation and the user redirected to the last active setting that occurred during the execution of the script.

In total, the greatest setback to the user accounts system is that every page that requires password protection must be written as a PHP file. The script is only provided in that language and is most commonly done so because of the connection between SQL. In either case, the extensive coding achieves the goal of limiting protection and incorporating user account systems to all PHP supported hosts through a simple “copy and paste” of the generated sample header.

MASS ELECTRONIC MAILING SYSTEM

The module that is most connected to the user accounts module—the mass electronic mailing system—has the most minimal script of the four modules. However, the mass mail system is much more limited in functionality. The sole purpose of this module is to send targeted e-mails to specific user groups previously defined in the user accounts installation.

Upon logging into this system, the user can only choose to e-mail the groups or send messages particular users. This function appeared to be easily integrated into the user account module as another function, however, because of the order of growth association with performing the mail option it was imperative that the two were separated. After the number of users surpasses one-hundred, the administrator experience a wait time for the code to be executed. Additionally, the mailing script does not require access to modify each individual user.

To allow administrators to customize the e-mail messages, HTML capabilities were integrated into both the administrative interface and background scripts. By simply adding a radio option in the form, a content determination conditional statement can be used as follow:

```

if ($format == "HTML") {
    // Always set content-type when sending HTML email
    $headers = "MIME-Version: 1.0" . "\r\n";
    $headers .= "Content-type:text/html;charset=iso-8859-1" . "\r\n";
    $headers .= "From: $from" . "\r\n";
    $headers .= "Reply-To: $from" . "\r\n";
    if ($cc != "") { $headers .= "Cc: $cc" . "\r\n"; }
    if ($bcc != "") { $headers .= "Bcc: $bcc" . "\r\n"; }
} else if ($format == "PLAIN") {
    $headers = "From: $from" . "\r\n";
    $headers .= "Reply-To: $from" . "\r\n";
    if ($cc != "") { $headers .= "Cc: $cc" . "\r\n"; }
    if ($bcc != "") { $headers .= "Bcc: $bcc" . "\r\n"; }
}

```

Close attention was used to separate and concatenate each line. The content-type was the critical component needed to send HTML messages. If provided for an e-mail sent in plain text, the entire message is shown as one big paragraph, regardless of spacing. Because of this, the default is set to Plain Text. This lack of headers makes it hard for the e-mail client to read the spacing entered into the textbox prior to sending the message.

At the conclusion of sending the e-mails, the results are generated and displayed for the administrator. The display shows the receivers of the e-mail, the sender, the subject, and the message itself. From here, a user can either send another e-mail, visit to the user accounts module, or simply log out of the system.

In this module, the most complicated piece is the pre-defined PHP mail() function. While the function is very simple in that it requires very little information in order to send an e-mail, it does require the necessary time to send each individual message. This limit becomes an issue when a particular user group grows very large. Because the mail() function, on average, takes just as much time as the standard e-mail client when sending messages, this module linearly becomes less effective with the growth rate of user accounts within a group.

CALENDAR AND EVENTS MANAGEMENT SYSTEM

The third module—the calendar and events management system—is one of the more dynamic modules. The coding for the calendar generates a visual that can be altered in color and size. The resulting calendar is dynamically created through a series of settings based on the current time, its relationship to the current day, month, and year, and properties associated with a calendar (i.e., seven days in a week, starting day of Sunday, etc.). To allow a user to integrate the calendar into any page, a reference to the include() function of PHP is needed. By including the supplied field on a PHP page, the page will incorporate the calendar based on the users specifications (Figure 3).

May 2011						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figure 3. Screen Shot of the Generated Calendar Script

To produce the code, many pre-defined functions within PHP are used. The key, however, is understanding how to manipulate the code to display the appropriate layout. To do so, a series of conditional statements are used as follows:

```

// determine variable associated with today's date
$date = time () ;
$today = date('d', $date);
$day_array = array("00", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12",
"13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
"30", "31");

// determine what month and year to show
if ($_GET['month'] == "") { $month = date('m', $date); } else { $month = $_GET['month']; }
if ($_GET['year'] == "") { $year = date('Y', $date); } else { $year = $_GET['year']; }

// determine the month variables
$ini_day = mktime(0, 0, 0, $month, 1, $year);
$ini_weekday = date('D', $ini_day) ;
$month_title = date('F', $ini_day);
$num_days = cal_days_in_month(0, $month, $year) ;

// determine the number of blank days prior to the first calendar day
switch($ini_weekday) {
    case "Sun": $blank = 0; break;
    case "Mon": $blank = 1; break;
    case "Tue": $blank = 2; break;
    case "Wed": $blank = 3; break;
    case "Thu": $blank = 4; break;
    case "Fri": $blank = 5; break;
    case "Sat": $blank = 6; break;
}

```

This code shows how dependent a calendar is on `date()` and `time()` functions (as expected). The day is the first variable determined in the script. An array is also established such that the string value that correlated to the number value can be acquired. To determine the month and year, the GET method is used in order to determine whether the viewer is requesting a particular month or year and if not, then to show the current month and year.

From there, the assignment of variables needed for calendar generation begins. The big pieces are determining the initial day of the month, what day of the month the selected month begins with, the name of the month, and the number of days within the month. Lastly, based on the initial day of the week, the code determines the number of blank boxes that have to be generated. Next, the table is written with the use of several constructing loops.

In order to display the events that are posted, the constructing loops have to determine the current day and select any item from the database that matches that date. This is most simply done by accessing the database within the conditional loop as show below.

```

$day_num = 01;
while ( $day_num <= $num_days )
{
    echo "<td class=\"default\">";
    echo "<table id=\"events\">";
    echo "<tr>";
    echo "<td class=\"number\"><p class=\"day_num\">$day_num</p></td>";
    echo "</tr><tr>";
    echo "<td class=\"event\"><p class=\"day_event\">";
    $day = $day_array[$day_num];
    $today_unix = mktime(0,0,0,$month,$day,$year);
    $tomorrow_unix = strtotime("+1 day", $today_unix);

    // query the database to call the operation
    $sql = "SELECT * FROM $tblname
          WHERE start_unix >= '$today_unix' AND start_unix <= '$tomorrow_unix'";
    $result = @mysql_query($sql,$connection) or die(mysql_error());

    while ($sql = mysql_fetch_object($result))
    {
        $title      = $sql -> title;
        $post       = $sql -> post;
        if ($post == "YES") { echo "$title"; }
    }

    echo "</p></td>";
    echo "</tr>";
    echo "</table>";
    echo "</td>";
    $day_num++;
    $day_count++;

    // make a new row after 7 days
    if ($day_count > 7) {
        echo "</tr><tr>";
        $day_count = 1;
    }
}

```

The loop defines the initial day and runs the loop until it matches the number of days in the month. Each block represents a table that includes the date number and the information (if

any) associated with that day. To pull the information, the current day and following day are used as the search limits and any events in the database that match the interval are displayed. Once the table is closed, the number of days is counted and if it equals seven, the loop resets and the row is closed. This action ensures that the calendar does not incorporate more than seven days on a single row.

Like the user accounts, the complexity of the calendar management system comes from the administrative control panel. In the control panel an administrator is able to add an event, modify an event, and delete, recover, or expunge an event.

To add an event requires minimal details, however, there are many pieces of information that can be incorporated. The form is supplied in a four-step process: general event information, supplemental highlights, event logistics, and posting information. The supplemental highlights information is not required at all and the posting information is only to inform the administrator of the event ID number and to decide whether the event is posted to the calendar now or later.

The important information is within the general information and the event logistics. Much of the information within step one inquires about the name of the event, its description, location, and time. The required information in step three inquires about the point of contact for the event. The most complicated part of the script comes in the “occurrence” input and the UNIX timestamp conversions (see “Miscellaneous Control Panel Scripts” – Appendix C).

The “occurrence” input allows the user to select whether an event occurs only once, daily, weekly, or yearly, and puts the number of events into to calendar based on the end date. The result is an individual event per interval provided between the start and end date. Because the timestamp is converted into the standard UNIX format, the methods makes use of seconds only (i.e., a day is added by adding $24*60*60$, a week by $7*24*60*60$, etc.) in relation to the first

day of the year. The use of seconds also makes it easier to export the year, month, and day, in addition to the hour, minutes, seconds, and meridiem of each event.

The modification of an event mirrors the add form exactly, with the exception of importing the data associated with the particular event selected and displaying it within the form. To choose an event to modify, the user is asked to search by year. This is used to reduce bandwidth, similar to the group selection for the mail function in the mass electronic mailing system. After editing the information, the event is then updated in the database. In order to view the event on the calendar, the event must be posted.

Like the user accounts, if the administrator wishes to delete an event, they may choose to do so at any time. The event may then be either recovered or expunged from the trash.

To make each event unique and identifiable within the database, each event is assigned a unique ID number. To reference the event, the only variable used is the ID number. In the event that the random number generated produces an identical number, the ID is reassigned and the program retries to enter the data. This method is most effective given that the probability of producing an identical event number is approximately $1e-10$.

Overall, the greatest set back with the calendar is also the requirement for PHP usage. However, the calendar allows for a higher level of customization and a more fluid integration into a website based solely on the user configuration files that define the visual components of the calendar.

FORM AND SURVEY CREATION SYSTEM

The last module is the most complex of them all—the forms and survey creation module. This module incorporates an extensive use of self-defined functions to execute the code. The

functions mirror the calendar module in that you can only add, modify, delete, recover, or expunge a form. The greatest difference comes from the limitations imposed once the database is created. Additionally, the data can only be read and constantly edited by incorporating an external source of information prior to generating the final form.

When a form is added, its name and ID number are stored into the database, however, all other information is used to produce a PHP file and a TXT file that stores all of the content until completion. To create the form, a custom function was defined as follows:

```
function addForm($idnum) {
    // gather the file paths
    $fileForm = "../forms/$idnum.php";
    $fileBackup = "../backup/$idnum.txt";
    // create the form file (.php)
    $fileFormHandle = @fopen($fileForm, "w") or exit("Unable to open file!");
    @fclose($fileFormHandle);
    // create the backup file (.txt)
    $fileBackupHandle = @fopen($fileBackup, "w") or exit("Unable to open file!");
    @fclose($fileBackupHandle);
}
```

The form is created using the pre-defined file handling functions. In this module, the modification algorithms are very different. The modification allows the administrator to add fields to the database until they choose to finalize the form. When adding a field to the form, the information is tracked using a long string that is separated with the “|” delimiter. This delimiter allows the information to be broken up into an array that can then be identified based of the data type. Each line is then added to the TXT file until the user is ready to finalize the form. To add to the file, another self-defined function is defined as follows:

```
function addFormConfig($idnum, $field_array) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "a") or exit("Unable to open file!");
    $stringData = "$field_array\r\n";
    @fwrite($fileBackupHandle, $stringData);
    @fclose($fileBackupHandle);
}
```

This action only interacts with the TXT file. The array is passed into the function and the function appends the information to the last line of the file and closes the file. This process is continued until all of the fields that the administrator wants are added. From there the user can then choose to go back and modify or delete any of the fields previously configured. The modification functions are also self-defined because of their complexity. For example, each line of code is a line within a text file. To get that line of code from the file, the following function is defined:

```
function getlineFormConfig($idnum, $field_id) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");
    $getlineConfig = "";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $data = explode("|", $line);

        if ($data[0] == $field_id) {
            $getlineConfig .= $line;
            break;
        }
    }
    @fclose($fileBackupHandle);
    return $getlineConfig;
}
```

The function requires a loop to read each line of the file. The file is read top-down until it reaches the line of code that matched the ID number associated with the chosen field. At that point it terminates the loop, closes the file, and returns the data. The data is then shown in a dynamically altered web page depending on the data type of the field. In order to accomplish this, the page requires heavy use of Javascript to make portions of the page visible under certain conditions. After the content has been edited, another self-defined modification function is used to locate the line of code in the file and replace it with the new line of code.

This process continues until the user is ready to post the form. Because of the use of several different data storage methods, there are certain stipulations associated with creating a form: questions on a form can be modified before or after the form is posted, however, the data type can no longer be modified; no fields can be added to a field once it has been finalized; and no field can be deleted once the form has been finalized.

These restrictions are the greatest setback associated with the forms module, however, the ability to just choose what input types a user wants on their site and to not have to learn how to write the PHP and SQL code alleviates all concerns associated with the limitations that arise due to the final posting of the form. As expected, however, these forms cannot be done without the use of PHP. In this module, the method of integration into the web page is done similarly to the calendar such that the include() function is the only needed script to display the form.

RESULTS OF ONLINE TESTING

After completing the modules, each was put to the test in an attempt to determine the true user capabilities and to greater understand the use of web applications for individuals with little to no experience in web scripting. To test the modules, the National Society of Black Engineers - MIT Chapter, Chocolate City, and the Office of Minority Education all utilized a different number of features.

To test the user account system, Chocolate City introduced a new brotherhood section of their website in order to log the individuals who have lived in the house both past an preset. The students there were able to install a modified version that also helps them to produce a “brothers” page by running a function that determines the groups of brothers based on class year. The

primary use, however, was to limit access of the web page to brothers only. The site now includes both functionalities without a complicated interface for the brothers to understand.

Currently, Chocolate City is also in the process of installing the calendar management system to incorporate their events into the website. At the moment, they have noted some difficulty in understanding the functionality of the integration of the calendar due to unclear instructions of the include() function. It is the hopes that this can be alleviated with a more automated integration process or a self-generating link similar to the methodology used for the events@MIT calendar.

The National Society of Black Engineers-MIT Chapter has begun to make heavy use of the mass electronic mailing system for their daily operations. To increase the capabilities, NSBE has allowed users to register on their website in order to limit access to the Chapter documents while also keeping track of those members who associate with the Chapter. They have defined their groups as corporate sponsors, collegiate students, pre-college students, graduate students, professors, and non-mit members. Their executive board suggests that the ease of categorization has helped them to better facilitate sending e-mails without having to remember each individual e-mail address for groups of people. They also believe that the increased personalization of the e-mails encouraged more people to come to some of their events within the past month.

Lastly, the Office of Minority Education just concluded it's testing of the forms creation module on their website. For the first time, they have produced an online form for their summer bridge program, Interphase, and as of Friday, they received one-hundred and fifty-one applications through the form. The application has helped them to reduce paper collection, has allowed them to reduce the amount of labor required to input the data into FileMaker Pro due to the auto-export feature of SQL, and has made possible the transition into other online

applications for programs like the Mentor Advocate Partnership and Laureates & Leaders. Additionally, they hope to expand the user accounts management system for their Industrial Advisory Council in order to provide password-protected sections for non-MIT entities that work with the office.

In order to allow self-testing during the entire process, an account was made for this thesis and allows for any person to access the administrative control panels self-hosted online currently. All modules are already pre-installed and capable of modifications. The account information is as follows:

Username: thesis_account
Password: meche2010

Website Links to Administrative Pages:

<http://www.jarrelljohnson.com/thesis/calendar/admin/>
<http://www.jarrelljohnson.com/thesis/events/admin/>
<http://www.jarrelljohnson.com/thesis/forms/admin/>
<http://www.jarrelljohnson.com/thesis/users/admin/>

CONCLUSION

Overall, it appears that the web modules do help individuals incorporate more intricate programming into their website, however, there are still some challenges that need to be addressed. More specifically, more care to explain how the modules are integrated after installation is needed for the calendar and form modules because they require more knowledge than just “copy and paste” like the user accounts module. Also, a recommendation was made to include individual messaging systems for mass mailing and a personalized control panel for each user within user accounts.

APPENDIX A – SCRIPTS FILE TREE

Below you will find a list of files associated with the scripts of the web modules. Each module serves as the high level folders within the list. The user accounts management system is named “users,” the calendar and events management module is named “calendar,” the mass electronic mailing module is named “massmail,” and the forms and survey creation module is named “forms.” All modules have an install and admin folder. Additionally some files are duplicated because they are needed for each module independently.

File Name

```
/calendar
  /admin
    admin-ctlpanel-bottom.php
    admin-ctlpanel-top.php
    admin-ctlpanel.html
    action-add.php
    action-addform.php
    action-del.php
    action-delget.php
    action-mod.php
    action-modform.php
    action-modget.php
    admin-ctlpanel.php
    admin-index.php
    index.php
  /install
    header.html
    install.html
    install_step1.php
    install_step2.php
    install_step3.php
    install_step4.php
  style-calendar.css
  style-forms.css
  style-main.css
  index.php
  script-calcreate.php
  script-logout.php
```



```
/forms
  /admin
    admin-ctlpanel-bottom.php
    admin-ctlpanel-top.php
    admin-ctlpanel.html
    action-add.php
    action-addform.php
    action-del.php
    action-delget.php
    action-finalize.php
    action-mod-addfield.php
    action-mod-addfieldform.php
    action-mod-modfield.php
    action-mod-modfieldform.php
    action-mod.php
    action-modform.php
    action-modget.php
    action-preview.php
    admin-ctlpanel.php
    admin-index.php
    index.php
  /backup
  /forms
  /install
    header.html
    install.html
    install_step1.php
    install_step2.php
    install_step3.php
    install_step4.php
  indep-forms.css
  style-forms.css
  style-main.css
  index.php
  forms-functn.php
  script-logout.php

/massmail
  /admin
    admin-ctlpanel-bottom.php
    admin-ctlpanel-top.php
    admin-ctlpanel.html
    action-email.php
    action-emailform.php
    action-emailget.php
    admin-ctlpanel.php
```

```
        admin-index.php
        index.php
/users
/install
        header.html
        install.html
        install_step1.php
style-forms.css
style-main.css
script-logout.php

/admin
        admin-ctlpanel-bottom.php
        admin-ctlpanel-top.php
        admin-ctlpanel.html
        action-add.php
        action-addform.php
        action-mod.php
        action-modform.php
        action-other.php
        action-otherform.php
        admin-ctlpanel.php
        admin-index.php
        index.php
/install
        header.html
        install.html
        install_step1.php
        install_step2.php
        install_step3.php
        install_admin.php
/loglogin
        login-log.html
        login-log.php
        index.php
style-forms.css
style-main.css
acct-activation.html
acct-banned.html
acct-changeemail.html
acct-changepass.html
acct-login.html
acct-loginerr.html
acct-noaccess.html
acct-notactivated.html
acct-register.html
```

acct-requestinfo.html
script-changeemail.php
script-changepass.php
script-logincheck.php
script-loglogin.php
script-logout.php
script-redirectation.php
script-register.php
script-requestinfo.php
index.php
users-functn.php
script-logout.php

For every folder, please note that the instillation process will add a users-config.php and users-config.txt file (with the exception of “forms” which will add a forms-config.php file and a forms-config.txt file). Additionally, user accounts will add a sample-header.php file.

APPENDIX B – INSTALLATION SCRIPT EXCERPTS

User Accounts Management System

/install/install.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Account Management | Installation</title>
  <link rel="stylesheet" type="text/css" href="../style-forms.css" />
  <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>

<body>

<div id="header">
  <h1 class="center white">Account Management Installation</h1>
</div>

<div id="content">
<p class="center red">Please read the <a href="../index.html">Instructions</a> prior to beginning the
installation.<br />&nbsp;</p>
<form method="POST" action="install_step1.php">

<table width="95%" align="center">
<tr>
  <td width="50%" valign="top">
    <fieldset class="install" style="width:500px;">
      <legend>Installation Information</legend>
      <ol><li>
        <label style="width:150px;">Database Name:</label>
        <input name="dbname" type="text" style="width:200px;"><span class="red"> *</span>
      </li><li>
        <label style="width:150px;">Database Server:</label>
        <input name="sqlserver" type="text" style="width:200px;"><span class="red"> *</span>
      </li><li>
        <label style="width:150px;">Database Username:</label>
        <input name="dbuser" type="text" style="width:200px;"><span class="red"> *</span>
      </li><li>
        <label style="width:150px;">Database Password:</label>
        <input name="dbpass" type="text" style="width:200px;"><span class="red"> *</span>
      </li><li>
        <label style="width:150px;">Install Directory:</label>
        <input name="directory" type="text" value="http://www.yoursite.com/users"
        style="width:200px;"><span class="red"> *</span>
      </li></ol>

```

```

<label style="width:150px;">Admin E-Mail:</label>
<input name="email_admin" type="text" value="webmaster@yoursite.com"
style="width:200px;"><span class="red"> *</span>
</li><li>
<label style="width:150px;">Time Zone:</label>
<select name="timezone" style="width:200px;">
  <option value="-12">-12</option>
  <option value="-11">-11</option>
  <option value="-10">-10</option>
  <option value="-9.5">-9.5</option>
  <option value="-9">-9</option>
  <option value="-8.5">-8.5</option>
  <option value="-8">-8 PST</option>
  <option value="-7">-7 MST</option>
  <option value="-6">-6 CMT</option>
  <option value="-5" selected>-5 EST</option>
  <option value="-4">-4 AST</option>
  <option value="-3.5">-3.5</option>
  <option value="-3">-3 ADT</option>
  <option value="-2">-2</option>
  <option value="-1">-1</option>
  <option value="00">00 GMT</option>
  <option value="+1">+1 CET</option>
  <option value="+2">+2</option>
  <option value="+3">+3</option>
  <option value="+3.5">+3.5</option>
  <option value="+4">+4</option>
  <option value="+4.5">+4.5</option>
  <option value="+5">+5</option>
  <option value="+5.5">+5.5</option>
  <option value="+6">+6</option>
  <option value="+6.5">+6.5</option>
  <option value="+7">+7</option>
  <option value="+8">+8</option>
  <option value="+9">+9</option>
  <option value="+9.5">+9.5</option>
  <option value="+10">+10</option>
  <option value="+10.5">+10.5</option>
  <option value="+11">+11</option>
  <option value="+12">+12</option>
  <option value="+13">+13</option>
  <option value="+14">+14</option>
</select><span class="red"> *</span>
</li></ol>
</fieldset>
</td>

<td width="50%" valign="top">
<fieldset class="install" style="width:500px;">
<legend>General Preferences</legend>
<ol><li>

```

```

<label style="width:150px;">Website Domain:</label>
<input name="domain" type="text" value="www.yoursite.com" style="width:200px;"><span
class="red"> *</span>
</li><li>
<label style="width:150px;">Redirection Link:</label>
<input name="redirect" type="text" value="http://www.yoursite.com/folder"
style="width:200px;"><span class="red"> *</span>
</li><li>
<label style="width:150px;">Require Verification?</label>
<select name="verified" style="width:200px;">
  <option value="YES" selected>No</option>
  <option value="NO">Yes</option>
</select><span class="red"> *</span>
</li><li>
<label style="width:150px;">Minimum Password Length:</label>
<select name="minlength" style="width:200px;">
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6" selected="selected">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
</select><span class="red"> *</span>
</li><li>
<label style="width:150px;">Maximum Password Length:</label>
<select name="maxlength" style="width:200px;">
  <option value="14">14</option>
  <option value="15">15</option>
  <option value="16" selected="selected">16</option>
  <option value="17">17</option>
  <option value="18">18</option>
  <option value="19">19</option>
  <option value="20">20</option>
</select><span class="red"> *</span>
</li><li>
<label style="width:150px;">Log Account Logins?</label>
<select name="loglogin" style="width:200px;">
  <option value="NO" selected>No</option>
  <option value="YES">Yes</option>
</select><span class="red"> *</span>
</li><li>
<label style="width:150px;">Number of User Groups?</label>
<select name="groupnum" style="width:200px;">
  <option value="0" selected>0</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>

```

```

        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        </select><span class="red"> *</span>
    </li></ol>
</fieldset>

<fieldset class="submit" style="width:500px;">
<ol><li>
<label style="width:150px;">&nbsp;</label>
<input name="submit" class="submit" type="submit" value="Continue Installation"
style="width:200px;"/>
</li></ol>
</fieldset>
</td>
</tr>
</table>
</form>
</div>

</body>
</html>

```

/install/index_step1.php

```
<?php
// archive the previous entries into the $_SESSION[] variables
session_start(install);

$_SESSION['dbname']           = $_POST['dbname'];
$_SESSION['sqlserver']        = $_POST['sqlserver'];
$_SESSION['dbuser']           = $_POST['dbuser'];
$_SESSION['dbpass']           = $_POST['dbpass'];
$_SESSION['tblname']          = "useraccounts";
$_SESSION['directory']        = $_POST['directory'];
$_SESSION['timezone']         = $_POST['timezone'];
$_SESSION['verified']         = $_POST['verified'];
$_SESSION['redirect']         = $_POST['redirect'];
$_SESSION['ctlpanel']         = $_POST['directory'] . "/acct-ctlpanel.html";
$_SESSION['minlength']        = $_POST['minlength'];
$_SESSION['maxlength']        = $_POST['maxlength'];
$_SESSION['loglogin']         = $_POST['loglogin'];
$_SESSION['groupnum']         = $_POST['groupnum'];
$_SESSION['domain']           = $_POST['domain'];
$_SESSION['email_admin']      = $_POST['email_admin'];

if (!$_POST['dbname'] || !$_POST['sqlserver'] || !$_POST['dbuser'] || !$_POST['dbpass'] ||
    !$_POST['timezone'] || $_POST['directory'] == "http://www.yoursite.com/users" || $_POST['domain'] ==
    "www.yoursite.com" || !$_POST['minlength'] || !$_POST['maxlength'] || $_POST['email_admin'] ==
    "webmaster@yoursite.com" || $_POST['redirect'] == "http://www.yoursite.com/folder") {
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";
    include("header.html");
    echo "<div id=\"content\"><p class=\"center red\">You must complete all of the fields, please <a
href=\"javascript:history.go(-1)\">Go Back</a> and complete all of the fields.</p></div>";
    exit;
}

if ($_POST['groupnum'] == "0") {
    header("Location:install_step2.php");
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Account Management | Installation</title>
    <link rel="stylesheet" type="text/css" href="../style-forms.css" />
    <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>

<body>
```



```

<div id="header">
    <h1 class="center white">Account Management Installation</h1>
</div>

<div id="content">
<p>&nbsp;</p>
<form method="POST" action="install_step2.php">
    <fieldset>
        <legend>Group Names</legend>
        <ol><li>
            <center>Before you can continue to Step 2, enter the names for your Groups.<br />
            (Do not use "Administrator" or "User" as a Group Name. They're pre-configured.)<br /></center>
        </li>
        <?php
            $i = 1;
            while ($i <= $_SESSION['groupnum'])
            {
                echo "<li>";
                echo "<label>Group $i</label>";
                echo "<input type='text' name='group$i'><span class='red'> *</span>";
                echo "</li>";
                $i++;
            }
        ?>
        </ol>
    </fieldset>

    <fieldset class="submit">
        <ol><li>
            <label>&nbsp;</label>
            <input name="submit" class="submit" type="submit" value="Proceed to Step 2" />
        </li></ol>
    </fieldset>
</form>
</div>

</body>
</html>

```

/install/install_step2.php

```
<?php
session_start(install);

echo "<link rel='stylesheet' type='text/css' href='../style-main.css'>";
echo "<link rel='stylesheet' type='text/css' href='../style-forms.css'>";
include('header.html');
echo "<div id='content'>";

$group_array = array();
$i = 1;
while ($i <= $_SESSION['groupnum'])
{
    $group = "group$i";
    array_push($group_array, "$_POST[$group]");
    $i++;
}

$i = 1;

// verify the SQL Server, the Database Username and Database Password
$conn = @mysql_connect($_SESSION['sqlserver'], $_SESSION['dbuser'], $_SESSION['dbpass'])
or die(mysql_error());
$db = @mysql_select_db($_SESSION['dbname'],$conn) or die(mysql_error());

// develop installation message
if ($db) {
    echo "<p class='center'>Connection to Database
<em><strong>$_SESSION[dbname]</strong></em> Successful.<br />";
} else {
    echo "<p class='center'>There was an error connecting to the database.<br />";
    echo "<a href='\"javascript:history.go(-2)\"'>Please go back and check your Database
information.</a></p>";
    exit;
}

// if databased is verified, begin to create the user table
$sql = "CREATE TABLE IF NOT EXISTS $_SESSION[tblname] (firstname VARCHAR(25), lastname
VARCHAR(25), username VARCHAR(25), password VARCHAR(50), usergroup
VARCHAR(20), userrights VARCHAR(20), changepass VARCHAR(1), email VARCHAR(100),
redirect VARCHAR(250), verified VARCHAR(3), last_login DATE)";

$result = @mysql_query($sql,$conn) or die(mysql_error());
if ($result) {
    echo "Table $_SESSION[tblname] has been created. ";
} else {
    echo "<p class='center'>There was an error creating the table.<br />";
    echo "<a href='\"javascript:history.go(-2)\"'>Please go back and check your
information.</a></p>";
    exit;
}
```

```

}

// continue to make a table for loglogin
$sql_log = "CREATE TABLE IF NOT EXISTS loglogin (username VARCHAR(25), login_date
VARCHAR(20), login_time VARCHAR(20), login_ip VARCHAR(20), opersys VARCHAR(30),
browser VARCHAR(30))";

$result_log = @mysql_query($sql_log,$connection) or die(mysql_error());

if ($result_log) {
    echo "Table loglogin has been created. ";
} else {
    echo "<p class='center'>There was an error creating the table.<br />";
    echo "<a href='\"javascript:history.go(-2)\"'>Please go back and check your
information.</a></p>";
    exit;
}

// continue to make a table for banned members
$sql_ban = "CREATE TABLE IF NOT EXISTS banned (no_access VARCHAR(30), type
VARCHAR(10))";

$result_ban = @mysql_query($sql_ban,$connection) or die(mysql_error());

if ($result_ban) {
    echo "Table banned has been created. ";
} else {
    echo "<p class='center'>There was an error creating the table.<br />";
    echo "<a href='\"javascript:history.go(-2)\"'>Please go back and check your
information.</a></p>";
    exit;
}

// continue to make a table for trash
$sql_trash = "CREATE TABLE IF NOT EXISTS trash (firstname VARCHAR(25), lastname
VARCHAR(25), username VARCHAR(25), password VARCHAR(50), usergroup VARCHAR(20),
userrights VARCHAR(20), changepass VARCHAR(1), email VARCHAR(100), redirect
VARCHAR(250), verified VARCHAR(3), last_login DATE, del_date DATE)";

$result_trash = @mysql_query($sql_trash,$connection) or die(mysql_error());

if ($result_trash) {
    echo "Table trash has been created.</p>";
} else {
    echo "<p class='center'>There was an error creating the table.<br />";
    echo "<a href='\"javascript:history.go(-2)\"'>Please go back and check your
information.</a></p>";
    exit;
}

echo "</div>";

```

```

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Account Management | Installation</title>
  <link rel="stylesheet" type="text/css" href="../style-forms.css" />
  <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>

<body>
<?php
$_SESSION['config'] = "<?php
// database name and table name
$dbname = \"$_SESSION[dbname]\";
$tblname = \"$_SESSION[tblname]\";

// database server name, username, and password
$sqlserver = \"$_SESSION[sqlserver]\";
$dbuser = \"$_SESSION[dbuser]\";
$dbpass = \"$_SESSION[dbpass]\";

// website domain
$domain = \"$_SESSION[domain]\";

// set to \"NO\" to disable LogLogin
$loglogin = \"$_SESSION[loglogin]\";

// location of files, ie http://www.yoursite.com/users
$directory = \"$_SESSION[directory]\";

// duration of cookie - (seconds*minutes*hours*days)
$duration = time()+(60*60*24*7);

// site administrator's email address
$email_admin = \"$_SESSION[email_admin]\";

// set time to Eastern Standard Time
$zone=3600*$_SESSION[timezone];

// verify new emails through email?
//yes = \"NO\" : no = \"YES\"
$verified = \"$_SESSION[verified]\";

// control panel location and redirect
$redirect = \"$_SESSION[redirect]\";
$ctlpanel = \"$_SESSION[ctlpanel]\";

// minimum and maximum password lengths

```

```

\ $minlength = $_SESSION[minlength];
\ $maxlength = $_SESSION[maxlength];

// array with the defined groups
\ $groupnum = $_SESSION[groupnum]+2;
\ $group_array = array("");
$si = 0;
while ($si < $_SESSION['groupnum'])
{
    $group = "group$si";
    $_SESSION['config'] .= "\"$group_array[$si]\", ";
    $si++;
}
$_SESSION['config'] .= "\"User\", ";
$_SESSION['config'] .= "\"Administrator\"";
?>";

// set the stored folder of the user account configurations
$stored_folder = getcwd();
$temp_folder = explode("/", $stored_folder);
array_pop($temp_folder);
$actual_folder = implode("/", $temp_folder);

$_SESSION['sampleheader'] = "<?php
// prevent caching of the web page
header("Expires: Sat, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: \".gmdate("D, d M Y H:i:s").\" GMT");
header("Cache-Control: post-check=0, pre-check=0\",false);
session_cache_limiter();

session_start();

// require the configuration and the functions files
require(" . $actual_folder . "/users-config.php");
require(" . $actual_folder . "/users-functn.php");

// check to see if the user is logged in
if (!$_SESSION['user_name']) {
    \ $_SESSION['temp_redirect'] = \"http://$domain\" . \$_SERVER['PHP_SELF'];
    header("Location: \" . $_SESSION['directory'] . "/acct-login.html");
    exit;
}
?>";

// create and write to file the users-config.text file
$FileName = "../users-config.txt";
$FileHandle = fopen($FileName, 'w') or die("Can't Open File $FileName");
$stringData = $_SESSION['config'];
fwrite($FileHandle, $stringData);
fclose($FileHandle);

```

```
// create and write to file the users-config.php file
$FileName = "./users-config.php";
$FileHandle = fopen($FileName, 'w') or die("Can't Open File $FileName");
$stringData = $_SESSION['config'];
fwrite($FileHandle, $stringData);
fclose($FileHandle);

// create and write to file the sample-header.php file
$FileName = "./sample-header.php";
$FileHandle = fopen($FileName, 'w') or die("Can't Open File $FileName");
$stringData = $_SESSION['sampleheader'];
fwrite($FileHandle, $stringData);
fclose($FileHandle);

include('install_admin.php');

?>
</div>

</body>
</html>
```

/install/install_admin.php

```
<?
session_start(install);
include('../users-config.php');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Account Management | Installation</title>
    <link rel="stylesheet" type="text/css" href="../style-forms.css" />
    <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>
<body>

<div id="header"><h1 class="center white">Account Management Installation</h1></div>

<div id="content"> <p>&nbsp;</p> <p>&nbsp;</p>
<form method="POST" action="install_step3.php">

    <fieldset>
    <legend>Create Your Administrator Account</legend>
    <ol><li>
    <label>First Name:</label>
    <input name="firstname" type="text" maxlength="50"><span class="red"> *</span>
    </li><li>
    <label>Last Name:</label>
    <input name="lastname" type="text" maxlength="50"><span class="red"> *</span>
    </li><li>
    <label>Username:</label>
    <input name="username" type="text" maxlength="50"><span class="red"> *</span>
    </li><li>
    <label>Password:</label>
    <input name="password" type="password" maxlength="25"><span class="red"> *</span>
    </li></ol>
    </fieldset>

    <fieldset class="submit">
    <ol><li>
    <label>&nbsp;</label>
    <input name="submit" class="submit" type="submit" value="Create Administrator" />
    </li></ol>
    </fieldset>
</form>
</div>

</body>
</html>
```

/install/install_step3.php

```
<?php
session_start(install);

include('header.html');
include('../users-functn.php');
include('../users-config.php');

$adminuser = $_POST['username'];

if (!$_POST['firstname'] || !$_POST['lastname'] || !$_POST['username'] || !$_POST['password']) {
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";
    echo "<div id=\"content\">";
    echo "<p class=\"center\">You must complete all of the fields, please <a
href=\"install_admin.php\">Go Back </a>and complete all of the fields.</p>";
    echo "</div>";
    exit;
}

$passcheck = AcceptPassword($_SESSION['minlength'], $_SESSION['maxlength'],
$_POST['password']);
if ($passcheck == "no") {
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";
    echo "<div id=\"content\">";
    echo "<p class=\"center\">You must use a password between $_SESSION[minlength] and
$_SESSION[maxlength] characters in length, please <a href=\"install_admin.php\">Go Back</a> and try
again.</p>";
    echo "</div>";
    exit;
}

// connect to the database to call the operation
$connection = @mysql_connect($_SESSION['sqlserver'], $_SESSION['dbuser'], $_SESSION['dbpass'])
or die(mysql_error());
$db = @mysql_select_db($_SESSION[dbname],$connection) or die(mysql_error());

$check_name = "SELECT * FROM $_SESSION[tblname] WHERE username = '$adminuser'";
$result_name = @mysql_query($check_name,$connection) or die(mysql_error());

// aquire the number of matching rows
$numrows = mysql_num_rows($result_name);

if ($numrows != 0) {
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
    echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";
    echo "<div id=\"content\">";
    echo "<p class=\"center\">That username already exists. Please <a
href=\"install_admin.php\">Go Back</a> and enter a different username.</p>";
    echo "</div>";
}
```



```

        exit;
    }

    $date = gmdate("Y-m-d");

    // add the administrator to the database
    $sql = "INSERT INTO $_SESSION[tablename] VALUES ('$_POST[firstname]', '$_POST[lastname]',
    '$_POST[username]', password($_POST[password]), 'User', 'Administrator', '0',
    '$_SESSION[email_admin]', '$_SESSION[redirect]', 'YES', '$date)";

    $result = @mysql_query($sql,$connection) or die(mysql_error());

    if($result)
    {
    ?>
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Account Management | Installation</title>
        <link rel="stylesheet" type="text/css" href="../style-forms.css" />
        <link rel="stylesheet" type="text/css" href="../style-main.css" />
    </head>

    <body>

    <div id="content">
    <?php
        echo "<table width=\"95%\" align=\"center\"><tr><td width=\"50%\" valign=\"top\">";
        echo "<textarea rows=\"50\" name=\"config\" style=\"width:100%;font-
        size:8pt;\">$_SESSION[config]</textarea>";
        echo "</td><td width=\"50%\" valign=\"top\">";
        echo "<p class=\"center\">The Administrator's Account was successfully created.<br />";
        echo "The configuration file was created as \"users-config.php\".<br />";
        echo "The back-up file was created as \"users-config.txt\".<br /></p>";
        echo "<p class=\"center\">The box to the right shows the code for configuration.<br />";
        echo "The box below shows the header to be placed on each PHP page.<br />&nbsp;";
        echo "<textarea rows=\"25\" name=\"config\" style=\"width:100%;font-
        size:8pt;\">$_SESSION[sampleheader]</textarea></p>";
        echo "<p class=\"center\">Please login as with the Administrator's Account to complete
        testing.<br />";
        echo "<a href =\"$_SESSION[directory]/acct-login.html\">Login Now</a><br />&nbsp;</p>";
        echo "</td></tr></table>";

    }
    ?>
    </div>

    </body>
    </html>

```

Mass Electronic Mailing System

/install/install.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>E-Mail Mailing | Installation</title>
  <link rel="stylesheet" type="text/css" href="../style-forms.css" />
  <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>

<body>

<div id="header">
  <h1 class="center white">E-Mail Mailing Installation</h1>
</div>

<div id="content">
<p class="center red">Please read the <a href="../index.html">Instructions</a> prior to beginning the
installation.<br />You <strong><em>MUST</em></strong> have the "User Accounts" module already
installed in order to use this module.<br />&nbsp;</p>

<form method="POST" action="install_step1.php">

  <fieldset>
    <legend>Installation Information</legend>
    <ol><li>
      <label>"User Accounts" Directory:</label>
      <input name="directory" type="text"><span class="red"> *</span>
    </li></ol>
  </fieldset>

  <fieldset class="submit">
    <ol><li>
      <label>&nbsp;</label>
      <input name="submit" class="submit" type="submit" value="Continue Installation" />
    </li></ol>
  </fieldset>
</form>

</div>

</body>
</html>
```

/install/install_step1.php

```
<?php
// archive the previous entries into the $_SESSION[] variables
session_start(install);

$_SESSION['directory']      = $_POST['directory'];

if (!$_POST['directory']) {
    echo "<link rel='stylesheet' type='text/css' href='../style-main.css'>";
    echo "<link rel='stylesheet' type='text/css' href='../style-forms.css'>";
    include('header.html');
    echo "<div id='content'><p class='center red'>You must enter the 'User Accounts' install
directory, ";
    echo "please <a href='javascript:history.go(-1)'>Go Back</a> and complete all of the
fields.</p></div>";
    exit;
}

$filename = $_SESSION['directory'] . "/users-config.php";
$sexists = @fopen($filename, "r");

if (!$sexists) {
    echo "<link rel='stylesheet' type='text/css' href='../style-main.css'>";
    echo "<link rel='stylesheet' type='text/css' href='../style-forms.css'>";
    include('header.html');
    echo "<div id='content'><p class='center red'>You must enter a valid 'User Accounts'
install directory, ";
    echo "please <a href='javascript:history.go(-1)'>Go Back</a> and complete all of the
fields.</p></div>";
    exit;
}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>E-Mail Mailing | Installation</title>
    <link rel="stylesheet" type="text/css" href="../style-forms.css" />
    <link rel="stylesheet" type="text/css" href="../style-main.css" />
</head>

<body>

<div id="header">
    <h1 class="center white">E-Mail Mailing Installation</h1>
</div>
```

```

<div id="content">
<p class="center">Copy operation for the configuration file in progress...</p>
<p class="center red">
<?php
// the name of the file wanted
$file = "users-config.php";

// properties for the source filename
$filename_array = parse_url($filename);
$filename_locat = $filename_array['path'];
$filename_parts = explode("/", $filename_locat);
$filename_fldr = count($filename_parts);

// properties for the destination filename
$dest_locat = $_SERVER['PHP_SELF'];
$dest_parts = explode("/", $dest_locat);
$dest_fldr = count($dest_parts);

// construct the correct filepath
$i = $filename_fldr - 1;
$j = $dest_fldr - 2;
$source_config = "";
$source_funcn = "";
// produce the correct number of abbreviations
while ($j > 0)
{
    if ($dest_parts[$j] == $filename_parts[$j]) { break; }
    $source_config .= "../";
    $source_funcn .= "../";
    $j--;
}
// append the remaining directories
$k = $j+1;
while ($k <= $i)
{
    $source_config .= $filename_parts[$k];
    $source_funcn .= $filename_parts[$k];
    if ($filename_parts[$k] == $file) { break; }
    $source_config .= "/";
    $source_funcn .= "/";
    $k++;
}

// get the contents from the source_config file
$configHandle = @fopen($source_config, "r");
$configContents = @file_get_contents($source_config);
fclose($configHandle);

// put the contents into the new config file
$configFile = "../users-config.php";
$configHandle = fopen($configFile, "w") or exit("Unable to open file!");

```

```

fwrite($configHandle, $configContents);
fclose($configHandle);

include($configFile);

// database name and table name
if ($dbname == "") { echo "Could not read the database Name.<br />";
    } else { echo "Aquired database name: <strong><em>$dbname</em></strong><br />"; }
if ($tblname == "") { echo "Could not read the table Name.<br />";
    } else { echo "Aquire table name: <em><strong>$tblname</strong></em><br />"; }

// database server name, username, and password
if ($sqlserver == "") { echo "Could not read the SQL server name.<br />";
    } else { echo "Determined the SQL server name.<br />"; }
if ($dbuser == "") { echo "Could not read the database username.<br />";
    } else { echo "Database username determined.<br />"; }
if ($dbpass == "") { echo "Could not read the database password.<br />";
    } else { echo "Database password determined.<br />"; }

// website domain
if ($domain == "") { echo "Could not read the domain.<br />";
    } else { echo "Aquired domain name: <em><strong>$domain</strong></em><br />"; }
if ($loglogin == "") { echo "Could not read the login properties.<br />";
    } else { echo "Determined login properties.<br />"; }

// location of files, ie http://www.yoursite.com/users
if ($directory == "") { echo "Could not read the directory.<br />";
    } else { echo "Aquired directory: <em><strong>$directory</strong></em><br />"; }

// duration of cookie - (seconds*minutes*hours*days)
if ($duration == "") { echo "Could not read the duration properties.<br />";
    } else { echo "Determined duration properties.<br />"; }

// site administrator's email address
if ($email_admin == "") { echo "Could not read the administrative e-mail.<br />";
    } else { echo "Aquired admin email: <em><strong>$email_admin</strong></em><br />"; }

// set time to Eastern Standard Time
if ($zone == "") { echo "Could not read the Time Zone properties.<br />";
    } else { echo "Determined Time Zone properties.<br />"; }

// verify new emails through email
if ($verified == "") { echo "Could not read the E-Mail properties.<br />";
    } else { echo "Determined E-mail properties.<br />"; }

// control panel location and redirect
if ($redirect == "") { echo "Could not read the redirection link.<br />";
    } else { echo "Aquired redirection: <em><strong>$redirect</strong></em><br />"; }
if ($ctlpanel == "") { echo "Could not read the control panel link.<br />";
    } else { echo "Aquired control panel: <em><strong>$ctlpanel</strong></em><br />"; }

```

```

// minimum and maximum password lengths
if ($minlength == "") { echo "Could not read the password properties.<br />";
    } else { echo "Determined password(min) properties.<br />"; }
if ($maxlength == "") { echo "Could not read the password properties.<br />";
    } else { echo "Determined password(max) properties.<br />"; }

// array with the defined groups
if ($groupnum == "") { echo "Could not read the group number properties.<br />";
    } else { echo "Determined number of usergroups.<br />"; }
if ($group_array == "") { echo "Could not read the group array properties.<br />";
    } else { echo "Determined types of usergroups.<br />"; }

```

```

?>
</p>

```

<p class="center">Copy operation completed...
Duplicating configuration file...</p>

```

<?php
// write the contents for the massmail users-config.php file
$newContents = "<?php
// database name and table name
\$dbname = \"\$dbname\";
\$tblname = \"\$tblname\";

// database server name, username, and password
\$sqlserver = \"\$sqlserver\";
\$dbuser = \"\$dbuser\";
\$dbpass = \"\$dbpass\";

// website domain
\$domain = \"\$domain\";

// set to \"NO\" to disable LogLogin
\$loglogin = \"\$loglogin\";

// location of files, ie http://www.yoursite.com/users
\$directory = \"\$directory\";

// duration of cookie - (seconds*minutes*hours*days)
\$duration = time()+(60*60*24*7);

// site administrator's email address
\$email_admin = \"\$email_admin\";

// set time to Eastern Standard Time
\$zone = $zone;

// verify new emails through email?
//yes = \"NO\" : no = \"YES\"
\$verified = \"\$verified\";

```

```

// control panel location and redirect
\$redirect = \"\$redirect\";
\$ctlpanel = \"\$ctlpanel\";

// minimum and maximum password lengths
\$minlength = $minlength;
\$maxlength = $maxlength;

// array with the defined groups
\$groupnum = $groupnum;
\$group_array = array("");
$i = 0;
while ($i < ($groupnum-2))
{
    $newContents .= "\"$group_array[$i]\", ";
    $i++;
}
$newContents .= "\"User\", ";
$newContents .= "\"Administrator\"";

?>;

// duplicate the contents into the new file
$newfile = "../users-config.php";
$newHandle = fopen($newfile, "w") or exit("Unable to open file!");
fwrite($newHandle, $newContents);
fclose($newHandle);
?>

<p class="center">Attempting to connect to the database...<br />
<?php
// connect to the database to call the operation
$connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
$db = @mysql_select_db($dbname,$connection) or die(mysql_error());

// develop installation message
if ($db) {
    echo "Connection to Database $dbname Successful...";
} else {
    echo "There was an error connecting to the database...<br />";
    echo "<a href=\"javascript:history.go(-1)\">Please go back and check your information
information.</a>";
    exit;
}

?>
</p>

<p class="center">The installation has been completed...<br />&nbsp;  <br />
<?php
echo "<center>";

```

```
echo "<form method=\"POST\" action=\"../admin/index.php\" target=\"display\">";
echo "<input type=\"button\" value=\"Complete Installation\" name=\"submit\"
onclick=\"window.location.href='../admin/index.php\" style=\"width:225px\">";
echo "</form>";
echo "</center>";
?>
</p>

</div>

</body>
</html>
```


APPENDIX C – MISCELLANEOUS CONTROL PANEL SCRIPTS

forms/forms-functn.php

```
<?php

// create the files for the form
function addForm($idnum) {
    // gather the file paths
    $fileForm = "../forms/$idnum.php";
    $fileBackup = "../backup/$idnum.txt";
    // create the form file (.php)
    $fileFormHandle = @fopen($fileForm, "w") or exit("Unable to open file!");
    @fclose($fileFormHandle);
    // create the backup file (.txt)
    $fileBackupHandle = @fopen($fileBackup, "w") or exit("Unable to open file!");
    @fclose($fileBackupHandle);
}

// delete the form and backup files
function deleteForm($idnum) {
    // gather the file paths
    $fileForm = "../forms/$idnum.php";
    $fileBackup = "../backup/$idnum.txt";
    $fileScripts = "../forms/$idnum-scripts.php";
    // delete the form file (.php)
    @unlink($fileForm);
    @unlink($fileScripts);
    // delete the backup file (.txt)
    @unlink($fileBackup);
}

// read the contents from the configuration text file
// and return the contents in a display for user viewing
function displayFormConfig($idnum) {
    global $sqlserver, $dbuser, $dbpass, $dbname, $tblname, $appname;

    echo "<br />";
    echo "<table width=\"100%\">";
    echo "<tr>";
        echo "<td width=\"50%\"><strong>Name</strong></td>";
        echo "<td width=\"20%\"><strong>Type</strong></td>";
        echo "<td width=\"10%\" align=\"center\"><strong>Modify?</strong></td>";
        echo "<td width=\"10%\" align=\"center\"><strong>Delete?</strong></td>";
        echo "<td width=\"10%\" align=\"center\"><strong>Require?</strong></td>";
    echo "</tr>";

    $filename = "../backup/$idnum.txt";
    $displayFile = @fopen($filename, "r");
```

```

// connect to the database to call the operation
$connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
$db = @mysql_select_db($dbname,$connection) or die(mysql_error());

// query the database to call the operation
$sql = "SELECT * FROM $tblname WHERE idnum ='$idnum'";
$result = @mysql_query($sql,$connection) or die(mysql_error());

// aquire the number of matching rows
$rownum = mysql_num_rows($result);

if ($rownum != 0) {
    while ($sql = mysql_fetch_object($result))
    {
        $finalized = $sql -> finalized;
    }
}

while(!feof($displayFile))
{
    $line = @fgets($displayFile);
    $data = explode("|", $line);

    if ($data[0] == "") { break; }

    // assign a new event id num
    if ($finalized == "NO") {
        echo "<tr><td colspan='5' style='border:solid #000;border-width:0 0 1px 0;'>&nbsp;</td></tr>";
        echo "<tr>";
        echo "<td>" . $data[1] . "</td>";
        echo "<td>" . $data[2] . "</td>";
        echo "<td align='center'>";
        echo "<a href='\"action-mod-modfieldform.php?idnum=$idnum&field_id=$data[0]\"'>";
            Confirm</a>";
        echo "</td>";
        echo "<td align='center'>";
        echo "<a href='\"action-mod-delfield.php?idnum=$idnum&field_id=$data[0]\"'>Confirm</a>";
        echo "</td>";
        echo "<td align='center'>" . $data[5] . "</td>";
        echo "</tr>";
    } else {
        echo "<tr>";
        echo "<td colspan='5' style='border:solid #000;border-width:0 0 1px 0;'>&nbsp;</td></tr>";
        echo "<tr>";
        echo "<td>" . $data[1] . "</td>";
        echo "<td>" . $data[2] . "</td>";
        echo "<td align='center'>";
        echo "<a href='\"action-mod-modfieldform.php?idnum=$idnum&field_id=$data[0]\"'>";
            Confirm</a>";
        echo "</td>";
    }
}

```

```

        echo "<td align=\"center\">" . "&nbsp;" . "</td>";
        echo "<td align=\"center\">" . $data[5] . "</td>";
        echo "</tr>";
    }
}

@fclose($displayFile);
echo "</table>";
echo "<p>&nbsp;</p>";
}

// get a line form the configuration files via action-mod-modfieldform.php
function getlineFormConfig($idnum, $field_id) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");
    $getlineConfig = "";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $data = explode("|", $line);

        if ($data[0] == $field_id) {
            $getlineConfig .= $line;
            break;
        }
    }
    @fclose($fileBackupHandle);
    return $getlineConfig;
}

// get a line form the configuration files via action-mod-modfieldform.php
function getPerLine($field_array) {
    // gather general information from post
    $field_id = $field_array[0];
    $field_name = $field_array[1];
    $datatype = $field_array[2];
    $field_text = $field_array[3];
    $field_help = $field_array[4];
    $field_required = $field_array[5];

    // gather line information based on datatype
    if ($datatype == "input") {
        // gather remaining values
        $input_length = $field_array[6];
        $input_type = $field_array[7];
        $input_pre = $field_array[8];
        $input_readonly = $field_array[9];

        return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
            'input_length', 'input_type', 'input_pre', 'input_readonly');
    }
}

```

```

} else if ($datatype == "textarea") {
    // gather remaining values
    $textarea_pre    = $field_array[6];
    $textarea_readonly = $field_array[7];

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
'textarea_pre', 'textarea_readonly');

} else if ($datatype == "radio") {
    // gather remaining values
    $radio_values    = $field_array[6];
    $radio_checked   = $field_array[7];

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
'radio_values', 'radio_checked');

} else if ($datatype == "checkbox") {
    $checkbox_values    = $field_array[6];
    $checkbox_checked   = $field_array[7];

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
'checkbox_values', 'checkbox_checked');

} else if ($datatype == "dropdown") {
    // gather remaining values
    $dropdown_values = $field_array[6];
    $dropdown_checked = $field_array[7];

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
'dropdown_values', 'dropdown_checked');

} else if ($datatype == "date") {
    // gather remaining values
    $year_from = $field_array[6];
    $year_to   = $field_array[7];

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required',
'year_from', 'year_to');
} else if ($datatype == "phone") {
    // no remaining values

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required');
} else if ($datatype == "address") {
    // no remaining values

    return compact('field_id', 'field_name', 'datatype', 'field_text', 'field_help', 'field_required');
}
}

// add to the form configuration files via action-mod-addfield.php

```

```

function addFormConfig($idnum, $field_array) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    // create the form file (.php)
    $fileBackupHandle = @fopen($fileBackup, "a") or exit("Unable to open file!");
    $stringData = "$field_array\r\n";
    @fwrite($fileBackupHandle, $stringData);
    echo "<span style=\"color:green;font-size:11px;\">
        The data has been written to the file...</span><br />";
    @fclose($fileBackupHandle);
    echo "<span style=\"color:green;font-size:11px;\">
        The field has been successfully added...</span><br />";
}

// delete form the configuration files via action-mod-delfield.php
function deleteFormConfig($idnum, $field_id) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");
    $scopiedConfig = "";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $data = explode("|", $line);

        if ($data[0] == $field_id) {

            continue;
        }

        $scopiedConfig .= $line;
    }
    @fclose($fileBackupHandle);
    $fileBackupHandle = @fopen($fileBackup, "w") or exit("Unable to open file!");
    @fwrite($fileBackupHandle, $scopiedConfig);
    echo "<span style=\"color:green;font-size:11px;\">
        The data has been re-written to the file...</span><br />";
    @fclose($fileBackupHandle);
    echo "<span style=\"color:green;font-size:11px;\">
        The field has been successfully updated...</span><br />";
}

// delete form the configuration files via action-mod-delfield.php
function modifyFormConfig($idnum, $field_id, $newfield_array) {
    // gather the file paths
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");
    $modlineConfig = "";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);

```

```

$data = explode("|", $line);

if ($data[0] == $field_id) {
    $modlineConfig .= "$newfield_array\r\n";
    echo "<span style=\"color:green;font-size:11px;\"> ";
    echo "<em><strong>$field_id</strong></em> has been modied...</span><br />";
    continue;
}

$modlineConfig .= $line;

}
@fclose($fileBackupHandle);

$fileBackupHandle = @fopen($fileBackup, "w") or exit("Unable to open file!");
@fwrite($fileBackupHandle, $modlineConfig);
echo "<span style=\"color:green;font-size:11px;\">
    The data has been re-written to the file...</span><br />";
@fclose($fileBackupHandle);
echo "<span style=\"color:green;font-size:11px;\">
    The field has been successfully updated...</span><br />";
}

// generate an id number per field of the form
function idnumField() {
    $letA = array("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m");
    $letB = array("n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z");
    $numA = rand(10,19);
    $numB = rand(100,999);
    $fieldID = $letA[rand(0,12)] . $numA . $letB[rand(0,12)] . $letA[rand(0,12)] . $numB;
    return $fieldID;
}

// functions to produce the actual form page
// both in preview mode and in create mode

// function to gather the true folder location
function getFolder() {
    // set the stored folder of the user account configurations
    $stored_folder = getcwd();
    $temp_folder = explode("/", $stored_folder);
    array_pop($temp_folder);
    $actual_folder = implode("/", $temp_folder);

    return $actual_folder;
}

// change form name to a lowercase connected string
function convertFormName($title) {
    $title = explode(" ", strtolower($title));
    $title = implode("_", $title);
}

```

```

return $title;
}

// function to add the header need to show the form
function addHeader($destination, $directory) {
    $header = "<?php\r\n";
    $header .= "\r\n";
    $header .= "// require the configuration files\r\n";
    $header .= "require('$destination/forms-config.php');\r\n";
    $header .= "require('$destination/forms-functn.php');\r\n";
    $header .= "\r\n";
    $header .= "// import the style sheets for the forms (text is the defaults of the site)\r\n";
    $header .= "echo \"<link rel=\\\"stylesheet\\\" type=\\\"text/css\\\"
        href=\\\"$directory/indep-forms.css\\\">\";\r\n";
    $header .= "?>\r\n";
    $header .= "\r\n";

    return $header;
}

// function to generate a preview of the form information
function previewFormContent($idnum, $form_name, $appname) {
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");

    $contents = "<?php\r\n";
    $contents .= "echo \"<br />\";\r\n";
    $contents .= "echo \"<form method=\\\"POST\\\" action=\\\"$idnum-scripts.php\\\"
        name=\\\"$appname\\\">\";\r\n";
    $contents .= "echo \"<fieldset>\";\r\n";
    $contents .= "echo \"<legend>$form_name</legend>\";\r\n";
    $contents .= "\r\n";

    $contents .= "echo \"<ol><li>\";\r\n";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $field_array = explode("|", $line);

        if ($line == "") { break; }

        @extract(getPerLine($field_array));

        $contents .= "echo \"<label class=\\\"default\\\">$field_text\";
            if ($field_required == "YES") { $contents .= "<span class=\\\"required\\\">*/span>"; }
            $contents .= "</label>\";\r\n";
            if ($field_help != "") { $contents .= "echo \"<label class=\\\"help\\\">$field_help</label>\";\r\n"; }
            if ($datatype == "input") {
                $contents .= "echo \"<input name=\\\"$field_id\\\" type=\\\"$input_type\\\"
                    value=\\\"$input_pre\\\"\";

```

```

if ($input_length != "") { $contents .= " maxlength=\\\\"$input_length\\\\""; }
if ($input_readonly == "YES") { $contents .= " readonly=\\\\"readonly\\\\""; }
$contents .= ">\\";\r\n";

} else if ($datatype == "textarea") {
    $contents .= "echo \\"<textarea name=\\\\"$field_id\\\\"";
    if ($textarea_readonly == "YES") { $contents .= " readonly=\\\\"readonly\\\\""; }
    $contents .= ">$textarea_pre</textarea>\\";\r\n";

} else if ($datatype == "radio") {
    $radio_values = explode(",", $radio_values);
    $num_radios = count($radio_values)-1;
    $contents .= "echo \\"<br />\\";\r\n";
    for ($counter = 0; $counter <= $num_radios; $counter += 1) {
        $contents .= "echo \\"</li><li>\\";\r\n";
        $contents .= "echo \\"<input name=\\\\"$field_id\\\\" class=\\\\"radio\\\\"";
        $contents .= " type=\\\\"$datatype\\\\" value=\\\\"$radio_values[$counter]\\\\"";
        if ($radio_checked == $radio_values[$counter]) { $contents .= " checked=\\\\"checked\\\\""; }
        $contents .= "> <span class=\\\\"radio\\\\">$radio_values[$counter]</span>\\";\r\n";
    }
} else if ($datatype == "checkbox") {
    $checkbox_values = explode(",", $checkbox_values);
    $num_boxes = count($checkbox_values)-1;
    $contents .= "echo \\"<br />\\";\r\n";
    for ($counter = 0; $counter <= $num_boxes; $counter += 1) {
        $contents .= "echo \\"</li><li>\\";\r\n";
        $contents .= "echo \\"<input name=\\\\"$field_id . \"$counter\\\\" class=\\\\"checkbox\\\\"";
        $contents .= " type=\\\\"$datatype\\\\" value=\\\\"$checkbox_values[$counter]\\\\"";
        if ($radio_checked == $checkbox_values[$counter]) { $contents .= " checked=\\\\"checked\\\\""; }
    }
}

$contents .= "> <span class=\\\\"radio\\\\">$checkbox_values[$counter]</span>\\";\r\n";
}

} else if ($datatype == "dropdown") {
    $dropdown_values = explode(",", $dropdown_values);
    $num_drops = count($dropdown_values)-1;
    $contents .= "echo \\"<select name=\\\\"$field_id\\\\">\\";\r\n";
    for ($counter = 0; $counter <= $num_drops; $counter += 1) {
        $contents .= "echo \\"<option value=\\\\"$dropdown_values[$counter]\\\\"";
        if ($dropdown_checked == $dropdown_values[$counter]) {
            $contents .= " selected=\\\\"selected\\\\""; }
        $contents .= ">$dropdown_values[$counter]</option>\\";\r\n";
    }
}
$contents .= "echo \\"</select> \\";\r\n";
} else if ($datatype == "date") {
    $contents .= "\$date = time ();\r\n";
    $contents .= "\$day = date('d', \$date);\r\n";
    $contents .= "\$month = date('m', \$date);\r\n";
    $contents .= "\$year = date('Y', \$date);\r\n";
    $contents .= "\r\n";
    $contents .= "?>\r\n";
}

```



```

$contentns .= "<select name=\"<?php echo \"\$field_id\" . \"year\"; ?>\" class=\"year\">\r\n";
$contentns .= "<?php\r\n";
$contentns .= "\$year_from = \$year_from;\r\n";
$contentns .= "\$year_to = \$year_to;\r\n";
$contentns .= "while (\$year_from <= \$year_to)";
$contentns .= "{\r\n";
$contentns .= "echo \"<option value=\\\"\\\"\$year_from\\\"\\\"\";";
$contentns .= "if (\$year_from == \$year) { echo \" selected=\\\"selected\\\"\"; }";
$contentns .= "echo \">\"</option>\";\r\n";
$contentns .= "\$year_from++;\r\n";
$contentns .= "}\r\n";
$contentns .= "?>\r\n";
$contentns .= "</select>\r\n";
$contentns .= "\r\n";

```

```

$contentns .= "<select name=\"<?php echo \"\$field_id\" . \"day\"; ?>\" class=\"day\">\r\n";
$contentns .= "<option value=\"01\"<?php if (\$day == \"01\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>01</option>\r\n";
$contentns .= "<option value=\"02\"<?php if (\$day == \"02\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>02</option>\r\n";
$contentns .= "<option value=\"03\"<?php if (\$day == \"03\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>03</option>\r\n";
$contentns .= "<option value=\"04\"<?php if (\$day == \"04\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>04</option>\r\n";
$contentns .= "<option value=\"05\"<?php if (\$day == \"05\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>05</option>\r\n";
$contentns .= "<option value=\"06\"<?php if (\$day == \"06\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>06</option>\r\n";
$contentns .= "<option value=\"07\"<?php if (\$day == \"07\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>07</option>\r\n";
$contentns .= "<option value=\"08\"<?php if (\$day == \"08\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>08</option>\r\n";
$contentns .= "<option value=\"09\"<?php if (\$day == \"09\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>09</option>\r\n";
$contentns .= "<option value=\"10\"<?php if (\$day == \"10\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>10</option>\r\n";
$contentns .= "<option value=\"11\"<?php if (\$day == \"11\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>11</option>\r\n";
$contentns .= "<option value=\"12\"<?php if (\$day == \"12\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>12</option>\r\n";
$contentns .= "<option value=\"13\"<?php if (\$day == \"13\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>13</option>\r\n";
$contentns .= "<option value=\"14\"<?php if (\$day == \"14\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>14</option>\r\n";
$contentns .= "<option value=\"15\"<?php if (\$day == \"15\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>15</option>\r\n";
$contentns .= "<option value=\"16\"<?php if (\$day == \"16\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>16</option>\r\n";
$contentns .= "<option value=\"17\"<?php if (\$day == \"17\") {
    echo \" selected=\\\"selected\\\"\"; } ?>>17</option>\r\n";
$contentns .= "<option value=\"18\"<?php if (\$day == \"18\") {

```

```

        echo \" selected=\\\"selected\\\""; } ?>>18</option>\r\n";
$contents .= "<option value='19'<?php if (\$day == '19') {
        echo \" selected=\\\"selected\\\""; } ?>>19</option>\r\n";
$contents .= "<option value='20'<?php if (\$day == '20') {
        echo \" selected=\\\"selected\\\""; } ?>>20</option>\r\n";
$contents .= "<option value='21'<?php if (\$day == '21') {
        echo \" selected=\\\"selected\\\""; } ?>>21</option>\r\n";
$contents .= "<option value='22'<?php if (\$day == '22') {
        echo \" selected=\\\"selected\\\""; } ?>>22</option>\r\n";
$contents .= "<option value='23'<?php if (\$day == '23') {
        echo \" selected=\\\"selected\\\""; } ?>>23</option>\r\n";
$contents .= "<option value='24'<?php if (\$day == '24') {
        echo \" selected=\\\"selected\\\""; } ?>>24</option>\r\n";
$contents .= "<option value='25'<?php if (\$day == '25') {
        echo \" selected=\\\"selected\\\""; } ?>>25</option>\r\n";
$contents .= "<option value='26'<?php if (\$day == '26') {
        echo \" selected=\\\"selected\\\""; } ?>>26</option>\r\n";
$contents .= "<option value='27'<?php if (\$day == '27') {
        echo \" selected=\\\"selected\\\""; } ?>>27</option>\r\n";
$contents .= "<option value='28'<?php if (\$day == '28') {
        echo \" selected=\\\"selected\\\""; } ?>>28</option>\r\n";
$contents .= "<option value='29'<?php if (\$day == '29') {
        echo \" selected=\\\"selected\\\""; } ?>>29</option>\r\n";
$contents .= "<option value='30'<?php if (\$day == '30') {
        echo \" selected=\\\"selected\\\""; } ?>>30</option>\r\n";
$contents .= "<option value='31'<?php if (\$day == '31') {
        echo \" selected=\\\"selected\\\""; } ?>>31</option>\r\n";
$contents .= "</select>\r\n";
$contents .= "\r\n";

$contents .= "<select name='<?php echo \"\$field_id\" . \"month\"; ?>' class='month'>\r\n";
$contents .= "<option value='01'<?php if (\$month == '01') {
        echo \" selected=\\\"selected\\\""; } ?>>January</option>\r\n";
$contents .= "<option value='02'<?php if (\$month == '02') {
        echo \" selected=\\\"selected\\\""; } ?>>February</option>\r\n";
$contents .= "<option value='03'<?php if (\$month == '03') {
        echo \" selected=\\\"selected\\\""; } ?>>March</option>\r\n";
$contents .= "<option value='04'<?php if (\$month == '04') {
        echo \" selected=\\\"selected\\\""; } ?>>April</option>\r\n";
$contents .= "<option value='05'<?php if (\$month == '05') {
        echo \" selected=\\\"selected\\\""; } ?>>May</option>\r\n";
$contents .= "<option value='06'<?php if (\$month == '06') {
        echo \" selected=\\\"selected\\\""; } ?>>June</option>\r\n";
$contents .= "<option value='07'<?php if (\$month == '07') {
        echo \" selected=\\\"selected\\\""; } ?>>July</option>\r\n";
$contents .= "<option value='08'<?php if (\$month == '08') {
        echo \" selected=\\\"selected\\\""; } ?>>August</option>\r\n";
$contents .= "<option value='09'<?php if (\$month == '09') {
        echo \" selected=\\\"selected\\\""; } ?>>September</option>\r\n";
$contents .= "<option value='10'<?php if (\$month == '10') {
        echo \" selected=\\\"selected\\\""; } ?>>October</option>\r\n";

```

```

$contentns .= "<option value=\"11\"<?php if (\$month == \"11\") {
    echo \" selected=\\\"selected\\\""; } ?>>November</option>\r\n";
$contentns .= "<option value=\"12\"<?php if (\$month == \"12\") {
    echo \" selected=\\\"selected\\\""; } ?>>December</option>\r\n";
$contentns .= "</select>\r\n";
$contentns .= "\r\n";

$contentns .= "<?php\r\n";
} else if ($datatype == "phone") {
    $contentns .= "?>\r\n";
    $contentns .= "<input name=\"<?php echo \"\$field_id\" . \"four\"; ?>\" class=\"fourdigits\">\r\n";
    $contentns .= "<input name=\"<?php echo \"\$field_id\" . \"three\"; ?>\" class=\"threedigits\">\r\n";
    $contentns .= "<input name=\"<?php echo \"\$field_id\" . \"areacode\"; ?>\"
        class=\"threedigits\">\r\n";
    $contentns .= "\r\n";

    $contentns .= "<?php\r\n";
} else if ($datatype == "address") {
    $contentns .= "?>\r\n";
    $contentns .= "<label class=\"default\">Address</label>\r\n";
    $contentns .= "<input name=\"<?php echo \"\$field_id\" . \"address\"; ?>\">\r\n";
    $contentns .= "<label class=\"default\">City</label>\r\n";
    $contentns .= "<input name=\"<?php echo \"\$field_id\" . \"city\"; ?>\">\r\n";
    $contentns .= "<label class=\"state\">State</label>\r\n";
    $contentns .= "<label class=\"zipcode\">Zip Code</label>\r\n";
    $contentns .= "<select name=\"<?php echo \"\$field_id\" . \"state\"; ?>\" class=\"state\">\r\n";
    $contentns .= "<option value=\"AL\">Alabama</option>\r\n";
    $contentns .= "<option value=\"AK\">Alaska</option>\r\n";
    $contentns .= "<option value=\"AZ\">Arizona</option>\r\n";
    $contentns .= "<option value=\"AR\">Arkansas</option>\r\n";
    $contentns .= "<option value=\"CA\">California</option>\r\n";
    $contentns .= "<option value=\"CO\">Colorado</option>\r\n";
    $contentns .= "<option value=\"CT\">Connecticut</option>\r\n";
    $contentns .= "<option value=\"DC\">District of Columbia</option>\r\n";
    $contentns .= "<option value=\"DE\">Delaware</option>\r\n";
    $contentns .= "<option value=\"FL\">Florida</option>\r\n";
    $contentns .= "<option value=\"GA\">Georgia</option>\r\n";
    $contentns .= "<option value=\"HI\">Hawaii</option>\r\n";
    $contentns .= "<option value=\"ID\">Idaho</option>\r\n";
    $contentns .= "<option value=\"IL\">Illinois</option>\r\n";
    $contentns .= "<option value=\"IN\">Indiana</option>\r\n";
    $contentns .= "<option value=\"IA\">Iowa</option>\r\n";
    $contentns .= "<option value=\"KS\">Kansas</option>\r\n";
    $contentns .= "<option value=\"KY\">Kentucky</option>\r\n";
    $contentns .= "<option value=\"LA\">Louisiana</option>\r\n";
    $contentns .= "<option value=\"ME\">Maine</option>\r\n";
    $contentns .= "<option value=\"MD\">Maryland</option>\r\n";
    $contentns .= "<option value=\"MA\">Massachusetts</option>\r\n";
    $contentns .= "<option value=\"MI\">Michigan</option>\r\n";
    $contentns .= "<option value=\"MN\">Minnesota</option>\r\n";
    $contentns .= "<option value=\"MS\">Mississippi</option>\r\n";

```

```

$contentents .= "<option value=\"MO\">Missouri</option>\r\n";
$contentents .= "<option value=\"MT\">Montana</option>\r\n";
$contentents .= "<option value=\"NE\">Nebraska</option>\r\n";
$contentents .= "<option value=\"NV\">Nevada</option>\r\n";
$contentents .= "<option value=\"NH\">New Hampshire</option>\r\n";
$contentents .= "<option value=\"NJ\">New Jersey</option>\r\n";
$contentents .= "<option value=\"NM\">New Mexico</option>\r\n";
$contentents .= "<option value=\"NY\">New York</option>\r\n";
$contentents .= "<option value=\"NC\">North Carolina</option>\r\n";
$contentents .= "<option value=\"ND\">North Dakota</option>\r\n";
$contentents .= "<option value=\"OH\">Ohio</option>\r\n";
$contentents .= "<option value=\"OK\">Oklahoma</option>\r\n";
$contentents .= "<option value=\"OR\">Oregon</option>\r\n";
$contentents .= "<option value=\"PA\">Pennsylvania</option>\r\n";
$contentents .= "<option value=\"RI\">Rhode Island</option>\r\n";
$contentents .= "<option value=\"SC\">South Carolina</option>\r\n";
$contentents .= "<option value=\"SD\">South Dakota</option>\r\n";
$contentents .= "<option value=\"TN\">Tennessee</option>\r\n";
$contentents .= "<option value=\"TX\">Texas</option>\r\n";
$contentents .= "<option value=\"UT\">Utah</option>\r\n";
$contentents .= "<option value=\"VT\">Vermont</option>\r\n";
$contentents .= "<option value=\"VA\">Virginia</option>\r\n";
$contentents .= "<option value=\"WA\">Washington</option>\r\n";
$contentents .= "<option value=\"WV\">West Virginia</option>\r\n";
$contentents .= "<option value=\"WI\">Wisconsin</option>\r\n";
$contentents .= "<option value=\"WY\">Wyoming</option>\r\n";
$contentents .= "</select>\r\n";
$contentents .= "<input name=\"<?php echo \"\$field_id\" . 'zipcode'\"; ?>\" class=\"zipcode\">\r\n";
$contentents .= "\r\n";

$contentents .= "<?php\r\n";
}

$contentents .= "echo \"</li><li>\";\r\n";

}

$contentents .= "?>\r\n";
$contentents .= "<br />\r\n";
$contentents .= "</li></ol>\r\n";
$contentents .= "\r\n";
$contentents .= "</fieldset>\r\n";
$contentents .= "\r\n";
$contentents .= "<fieldset class=\"submit\">\r\n";
$contentents .= "<ol><li>\r\n";
$contentents .= "<label>&nbsp;</label>\r\n";
$contentents .= "<input class=\"submit\" type=\"button\" value=\"Return\" onclick=\"history.go(-2)\"
/>\r\n";
$contentents .= "</li></ol>\r\n";
$contentents .= "</fieldset>\r\n";
$contentents .= "</form>\r\n";

```

```

@fclose($fileBackupHandle);

return $contents;
}

// function to finalize the form information
function finalizeFormContent($idnum, $form_name, $appname, $directory) {
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");

    $contents = "<?php\r\n";
    $contents .= "echo \"<br />\";\r\n";
    $contents .= "echo \"<form method=\\\\"POST\\\\" action=\\\\"$directory/forms/$idnum-scripts.php\\\\"
        name=\\\\"$appname\\\\">\";\r\n";
    $contents .= "echo \"<fieldset>\";\r\n";
    $contents .= "echo \"<legend>$form_name</legend>\";\r\n";
    $contents .= "\r\n";

    $contents .= "echo \"<ol><li>\";\r\n";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $field_array = explode("|", $line);

        if ($line == "") { break; }
        @extract(getPerLine($field_array));

        $contents .= "echo \"<label class=\\\\"default\\\\">$field_text\";
        if ($field_required == "YES") { $contents .= "<span class=\\\\"required\\\\">*/</span>\"; }
        $contents .= "</label>\";\r\n";
        if ($field_help != "") { $contents .= "echo \"<label class=\\\\"help\\\\">$field_help</label>\";\r\n"; }
        if ($datatype == "input") {
            $contents .= "echo \"<input name=\\\\"$field_id\\\\" type=\\\\"$input_type\\\\"
                value=\\\\"$input_pre\\\\"\";
            if ($input_length != "") { $contents .= "maxlength=\\\\"$input_length\\\\"\"; }
            if ($input_readonly == "YES") { $contents .= "readonly=\\\\"readonly\\\\"\"; }
            $contents .= ">\";\r\n";
        }
        } else if ($datatype == "textarea") {
            $contents .= "echo \"<textarea name=\\\\"$field_id\\\\"\";
            if ($textarea_readonly == "YES") { $contents .= "readonly=\\\\"readonly\\\\"\"; }
            $contents .= ">$textarea_pre</textarea>\";\r\n";
        }
        } else if ($datatype == "radio") {
            $radio_values = explode(",", $radio_values);
            $num_radios = count($radio_values)-1;
            $contents .= "echo \"<br />\";\r\n";
            for ($counter = 0; $counter <= $num_radios; $counter += 1) {
                $contents .= "echo \"</li><li>\";\r\n";
                $contents .= "echo \"<input name=\\\\"$field_id\\\\" class=\\\\"radio\\\\"\";
                $contents .= " type=\\\\"$datatype\\\\" value=\\\\"$radio_values[$counter]\\\\"\";
            }
        }
    }
}

```

```

        if ($radio_checked == $radio_values[$counter]) { $contents .= " checked=\\\\"checked\\\\""; }
        $contents .= "> <span class=\\\\"radio\\\\">$radio_values[$counter]</span>\\\\";\\n";
    }
} else if ($datatype == "checkbox") {
    $checkbox_values = explode(",", $checkbox_values);
    $num_boxes = count($checkbox_values)-1;
    $contents .= "echo \\<br />\\\\";\\n";
    for ($counter = 0; $counter <= $num_boxes; $counter += 1) {
        $contents .= "echo \\</li><li>\\\\";\\n";
        $contents .= "echo \\<input name=\\\\"$field_id" . "$counter\\\\" class=\\\\"checkbox\\\\"";
        $contents .= " type=\\\\"$datatype\\\\" value=\\\\"$checkbox_values[$counter]\\\\"";
        if ($radio_checked == $checkbox_values[$counter]) { $contents .= " checked=\\\\"checked\\\\""; }
    }
    $contents .= "> <span class=\\\\"radio\\\\">$checkbox_values[$counter]</span>\\\\";\\n";
}
} else if ($datatype == "dropdown") {
    $dropdown_values = explode(",", $dropdown_values);
    $num_drops = count($dropdown_values)-1;
    $contents .= "echo \\<select name=\\\\"$field_id\\\\">\\\\";\\n";
    for ($counter = 0; $counter <= $num_drops; $counter += 1) {
        $contents .= "echo \\<option value=\\\\"$dropdown_values[$counter]\\\\"";
        if ($dropdown_checked == $dropdown_values[$counter]) {
            $contents .= " selected=\\\\"selected\\\\""; }
        $contents .= ">$dropdown_values[$counter]</option>\\\\";\\n";
    }
    $contents .= "echo \\</select> \\\\";\\n";
} else if ($datatype == "date") {
    $contents .= "\\$date = time ();\\n";
    $contents .= "\\$day = date('d', \\$date);\\n";
    $contents .= "\\$month = date('m', \\$date);\\n";
    $contents .= "\\$year = date('Y', \\$date);\\n";
    $contents .= "\\n";
    $contents .= "?>\\n";

    $contents .= "<select name=\\\\"<?php echo '\\$field_id' . 'year\\'; ?>\\\\" class=\\\\"year\\\\">\\n";
    $contents .= "<?php\\n";
    $contents .= "\\$year_from = $year_from;\\n";
    $contents .= "\\$year_to = $year_to;\\n";
    $contents .= "while (\\$year_from <= $year_to)";
    $contents .= "{\\n";
    $contents .= "echo \\<option value=\\\\"\\$year_from\\\\"";
    $contents .= "if (\\$year_from == \\$year) { echo \\ selected=\\\\"selected\\\\""; }";
    $contents .= "echo \\>\\$year_from</option>\\\\";\\n";
    $contents .= "\\$year_from++;\\n";
    $contents .= "}\\n";
    $contents .= "?>\\n";
    $contents .= "</select>\\n";
    $contents .= "\\n";

    $contents .= "<select name=\\\\"<?php echo '\\$field_id' . 'day\\'; ?>\\\\" class=\\\\"day\\\\">\\n";
    $contents .= "<select name=\\\\"<?php echo '\\$field_id' . 'day\\'; ?>\\\\" class=\\\\"day\\\\">\\n";

```



```

        echo \" selected=\\\"selected\\\""; } ?>>26</option>\r\n";
$contents .= "<option value=\"27\"<?php if (\$day == \"27\") {
        echo \" selected=\\\"selected\\\""; } ?>>27</option>\r\n";
$contents .= "<option value=\"28\"<?php if (\$day == \"28\") {
        echo \" selected=\\\"selected\\\""; } ?>>28</option>\r\n";
$contents .= "<option value=\"29\"<?php if (\$day == \"29\") {
        echo \" selected=\\\"selected\\\""; } ?>>29</option>\r\n";
$contents .= "<option value=\"30\"<?php if (\$day == \"30\") {
        echo \" selected=\\\"selected\\\""; } ?>>30</option>\r\n";
$contents .= "<option value=\"31\"<?php if (\$day == \"31\") {
        echo \" selected=\\\"selected\\\""; } ?>>31</option>\r\n";
$contents .= "</select>\r\n";
$contents .= "\r\n";

$contents .= "<select name=\"<?php echo \"\$field_id\" . \"month\"; ?>\" class=\"month\">\r\n";
$contents .= "<option value=\"01\"<?php if (\$month == \"01\") {
        echo \" selected=\\\"selected\\\""; } ?>>January</option>\r\n";
$contents .= "<option value=\"02\"<?php if (\$month == \"02\") {
        echo \" selected=\\\"selected\\\""; } ?>>February</option>\r\n";
$contents .= "<option value=\"03\"<?php if (\$month == \"03\") {
        echo \" selected=\\\"selected\\\""; } ?>>March</option>\r\n";
$contents .= "<option value=\"04\"<?php if (\$month == \"04\") {
        echo \" selected=\\\"selected\\\""; } ?>>April</option>\r\n";
$contents .= "<option value=\"05\"<?php if (\$month == \"05\") {
        echo \" selected=\\\"selected\\\""; } ?>>May</option>\r\n";
$contents .= "<option value=\"06\"<?php if (\$month == \"06\") {
        echo \" selected=\\\"selected\\\""; } ?>>June</option>\r\n";
$contents .= "<option value=\"07\"<?php if (\$month == \"07\") {
        echo \" selected=\\\"selected\\\""; } ?>>July</option>\r\n";
$contents .= "<option value=\"08\"<?php if (\$month == \"08\") {
        echo \" selected=\\\"selected\\\""; } ?>>August</option>\r\n";
$contents .= "<option value=\"09\"<?php if (\$month == \"09\") {
        echo \" selected=\\\"selected\\\""; } ?>>September</option>\r\n";
$contents .= "<option value=\"10\"<?php if (\$month == \"10\") {
        echo \" selected=\\\"selected\\\""; } ?>>October</option>\r\n";
$contents .= "<option value=\"11\"<?php if (\$month == \"11\") {
        echo \" selected=\\\"selected\\\""; } ?>>November</option>\r\n";
$contents .= "<option value=\"12\"<?php if (\$month == \"12\") {
        echo \" selected=\\\"selected\\\""; } ?>>December</option>\r\n";
$contents .= "</select>\r\n";
$contents .= "\r\n";
$contents .= "<?php\r\n";
} else if ($datatype == "phone") {
    $contents .= "?>\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"four\"; ?>\" class=\"fourdigits\">\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"three\"; ?>\" class=\"threedigits\">\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"areacode\"; ?>\"
        class=\"threedigits\">\r\n";
    $contents .= "\r\n";

$contents .= "<?php\r\n";

```



```

} else if ($datatype == "address") {
    $contents .= "?>\r\n";
    $contents .= "<label class=\"default\">Address</label>\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"address\"; ?>\">\r\n";
    $contents .= "<label class=\"default\">City</label>\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"city\"; ?>\">\r\n";
    $contents .= "<label class=\"state\">State</label>\r\n";
    $contents .= "<label class=\"zipcode\">Zip Code</label>\r\n";
    $contents .= "<select name=\"<?php echo \"\$field_id\" . \"state\"; ?>\" class=\"state\">\r\n";
    $contents .= "<option value=\"AL\">Alabama</option>\r\n";
    $contents .= "<option value=\"AK\">Alaska</option>\r\n";
    $contents .= "<option value=\"AZ\">Arizona</option>\r\n";
    $contents .= "<option value=\"AR\">Arkansas</option>\r\n";
    $contents .= "<option value=\"CA\">California</option>\r\n";
    $contents .= "<option value=\"CO\">Colorado</option>\r\n";
    $contents .= "<option value=\"CT\">Connecticut</option>\r\n";
    $contents .= "<option value=\"DC\">District of Columbia</option>\r\n";
    $contents .= "<option value=\"DE\">Delaware</option>\r\n";
    $contents .= "<option value=\"FL\">Florida</option>\r\n";
    $contents .= "<option value=\"GA\">Georgia</option>\r\n";
    $contents .= "<option value=\"HI\">Hawaii</option>\r\n";
    $contents .= "<option value=\"ID\">Idaho</option>\r\n";
    $contents .= "<option value=\"IL\">Illinois</option>\r\n";
    $contents .= "<option value=\"IN\">Indiana</option>\r\n";
    $contents .= "<option value=\"IA\">Iowa</option>\r\n";
    $contents .= "<option value=\"KS\">Kansas</option>\r\n";
    $contents .= "<option value=\"KY\">Kentucky</option>\r\n";
    $contents .= "<option value=\"LA\">Louisiana</option>\r\n";
    $contents .= "<option value=\"ME\">Maine</option>\r\n";
    $contents .= "<option value=\"MD\">Maryland</option>\r\n";
    $contents .= "<option value=\"MA\">Massachusetts</option>\r\n";
    $contents .= "<option value=\"MI\">Michigan</option>\r\n";
    $contents .= "<option value=\"MN\">Minnesota</option>\r\n";
    $contents .= "<option value=\"MS\">Mississippi</option>\r\n";
    $contents .= "<option value=\"MO\">Missouri</option>\r\n";
    $contents .= "<option value=\"MT\">Montana</option>\r\n";
    $contents .= "<option value=\"NE\">Nebraska</option>\r\n";
    $contents .= "<option value=\"NV\">Nevada</option>\r\n";
    $contents .= "<option value=\"NH\">New Hampshire</option>\r\n";
    $contents .= "<option value=\"NJ\">New Jersey</option>\r\n";
    $contents .= "<option value=\"NM\">New Mexico</option>\r\n";
    $contents .= "<option value=\"NY\">New York</option>\r\n";
    $contents .= "<option value=\"NC\">North Carolina</option>\r\n";
    $contents .= "<option value=\"ND\">North Dakota</option>\r\n";
    $contents .= "<option value=\"OH\">Ohio</option>\r\n";
    $contents .= "<option value=\"OK\">Oklahoma</option>\r\n";
    $contents .= "<option value=\"OR\">Oregon</option>\r\n";
    $contents .= "<option value=\"PA\">Pennsylvania</option>\r\n";
    $contents .= "<option value=\"RI\">Rhode Island</option>\r\n";
    $contents .= "<option value=\"SC\">South Carolina</option>\r\n";
    $contents .= "<option value=\"SD\">South Dakota</option>\r\n";

```

```

    $contents .= "<option value=\"TN\">Tennessee</option>\r\n";
    $contents .= "<option value=\"TX\">Texas</option>\r\n";
    $contents .= "<option value=\"UT\">Utah</option>\r\n";
    $contents .= "<option value=\"VT\">Vermont</option>\r\n";
    $contents .= "<option value=\"VA\">Virginia</option>\r\n";
    $contents .= "<option value=\"WA\">Washington</option>\r\n";
    $contents .= "<option value=\"WV\">West Virginia</option>\r\n";
    $contents .= "<option value=\"WI\">Wisconsin</option>\r\n";
    $contents .= "<option value=\"WY\">Wyoming</option>\r\n";
    $contents .= "</select>\r\n";
    $contents .= "<input name=\"<?php echo \"\$field_id\" . \"zipcode\"; ?>\" class=\"zipcode\">\r\n";
    $contents .= "\r\n";

    $contents .= "<?php\r\n";
}

$contents .= "echo \"</li><li>\";\r\n";

}
$contents .= "?>\r\n";
$contents .= "<br />\r\n";
$contents .= "</li></ol>\r\n";
$contents .= "\r\n";
$contents .= "</fieldset>\r\n";
$contents .= "\r\n";
$contents .= "<fieldset class=\"submit\">\r\n";
$contents .= "<ol><li>\r\n";
$contents .= "<label>&nbsp;</label>\r\n";
$contents .= "<input class=\"submit\" type=\"reset\" value=\"Reset\" />\r\n";
$contents .= "<input class=\"submit\" type=\"submit\" value=\"Submit\" />\r\n";
$contents .= "</li></ol>\r\n";
$contents .= "</fieldset>\r\n";
$contents .= "</form>\r\n";
@fclose($fileBackupHandle);

return $contents;
}

// develop the scripts necessary to write to the form database
function finalizeFormScripts($idnum, $form_name, $appname, $destination, $directory, $domain) {
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");

    $contents = "<?php\r\n";
    $contents .= "// prevent caching of the web page\r\n";
    $contents .= "header(\"Expires: Sat, 01 Jan 2000 00:00:00 GMT\");\r\n";
    $contents .= "header(\"Last-Modified: \" . gmdate(\"D, d M Y H:i:s\") . \" GMT\");\r\n";
    $contents .= "header(\"Cache-Control: post-check=0, pre-check=0\",false);\r\n";
    $contents .= "session_cache_limiter();\r\n";
    $contents .= "\r\n";
    $contents .= "session_start();\r\n";

```

```

$contentns .= "\r\n";
$contentns .= "// require the configuration files\r\n";
$contentns .= "require('$destination/forms-config.php');\r\n";
$contentns .= "require('$destination/forms-functn.php');\r\n";
$contentns .= "\r\n";

$contentns .= "\$msg = \"\";\r\n";
$contentns .= "\$error = 0;\r\n";

$values = "";
while(!feof($fileBackupHandle))
{
$line = @fgets($fileBackupHandle);
$field_array = explode("|", $line);

if ($line == "") { break; }
@extract(getPerLine($field_array));

if ($datatype == "input") {
$contentns .= "\$field_name = stripslashes(preg_replace('/^/', '\\', \$POST['$field_id']));\r\n";
if ($field_required == "YES") {
$contentns .= "if (\$field_name == \$input_pre) {
$msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \$error++;
}";
}
$contentns .= "\r\n";
$values .= "\$field_name, ";
} else if ($datatype == "textarea") {
$contentns .= "\$field_name = stripslashes(preg_replace('/^/', '\\', \$POST['$field_id']));\r\n";
if ($field_required == "YES") {
$contentns .= "if (\$field_name == \$textarea_pre) {
$msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \$error++;
}";
}
$contentns .= "\r\n";
$values .= "\$field_name, ";
} else if ($datatype == "radio") {
$contentns .= "\$field_name = stripslashes(preg_replace('/^/', '\\', \$POST['$field_id']));\r\n";
if ($field_required == "YES") {
$contentns .= "if (\$field_name == \"\") {
$msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \$error++;
}";
}
$contentns .= "\r\n";
$values .= "\$field_name, ";
} else if ($datatype == "checkbox") {
$checkbox_values = explode(",", $checkbox_values);
$num_boxes = count($checkbox_values)-1;
$checkboxrequirement = "";
$checkboxval = "\"";
for ($counter = 0; $counter <= $num_boxes; $counter += 1) {

```

```

$content .= "\$field_name";
$content .= "$counter = stripslashes(preg_replace('/^', '\\', \$_POST['$field_id'];
$content .= "$counter']));\r\n";
$checkboxval .= "\$field_name" . "$counter,";
if ($counter != $num_boxes) {
    $checkboxrequirement .= ("\$field_name" . "$counter == '\\') && "; }
else { $checkboxrequirement .= ("\$field_name" . "$counter == '\\"); }
}
$checkboxval .= "\";";
$content .= "\$field_name = $checkboxval\r\n";
if ($field_required == "YES") {
    $content .= "if ($checkboxrequirement) {
        \msg .= \"You must enter a value for the field <b>$field_name</b><br>\"; \error++;
    }";
}
$content .= "\r\n";
$values .= "\$field_name', ";
} else if ($datatype == "dropdown") {
    $content .= "\$field_name = stripslashes(preg_replace('/^', '\\', \$_POST['$field_id']));\r\n";
    if ($field_required == "YES") {
        $content .= "if (\$field_name == '\\') {
            \msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \error++;
        }";
    }
}
$content .= "\r\n";
$values .= "\$field_name', ";
} else if ($datatype == "date") {
    $content .= "\$field_name" . "month =
        stripslashes(preg_replace('/^', '\\', \$_POST['$field_id' . 'month']));\r\n";
    $content .= "\$field_name" . "day =
        stripslashes(preg_replace('/^', '\\', \$_POST['$field_id' . 'day']));\r\n";
    $content .= "\$field_name" . "year =
        stripslashes(preg_replace('/^', '\\', \$_POST['$field_id' . 'year']));\r\n";
    $content .= "\$field_name = \"\$field_name\" . "year-\$field_name" . "month-\$field_name" .
"day\";\r\n";
    $content .= "\r\n";
    $values .= "\$field_name', ";
} else if ($datatype == "phone") {
    $content .= "\$field_name = \"\$_POST[$field_id' . "areacode]-\$_POST[$field_id' . "three]-
\$_POST[$field_id' . "four]\";\r\n";
    if ($field_required == "YES") {
        $content .= "if (\$field_name == \"--\") {
            \msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \error++;
        }";
    }
}
$content .= "\r\n";
$values .= "\$field_name', ";
} else if ($datatype == "address") {
    $content .= "\$field_name" . "address =
        stripslashes(preg_replace('/^', '\\', \$_POST['$field_id' . 'address']));\r\n";
    $content .= "\$field_name" . "city =

```

```

stripslashes(preg_replace("/\/\^", "\\", \$_POST['$field_id' . "city']));\r\n";
$content .= "\$$field_name" . "state =
stripslashes(preg_replace("/\/\^", "\\", \$_POST['$field_id' . "state']));\r\n";
$content .= "\$$field_name" . "zipcode =
stripslashes(preg_replace("/\/\^", "\\", \$_POST['$field_id' . "zipcode']));\r\n";
$addressrequirement = "(\$$field_name" . "address == \"\") || (\$$field_name" . "city == \"\") || ";
$addressrequirement .= "(\$$field_name" . "state == \"\") || (\$$field_name" . "zipcode == \"\")";
if ($field_required == "YES") {
    $content .= "if ($addressrequirement) {
        \msg .= \"You must enter a value for the field <b>$field_name</b><br />\"; \Error++;
    }";
}
$content .= "\r\n";
$values .= "\$$field_name" . "address' ";
$values .= "\$$field_name" . "city' ";
$values .= "\$$field_name" . "state' ";
$values .= "\$$field_name" . "zipcode' ";
}

$content .= "\r\n";

}
$content .= "\$date = date('Y-m-d');\r\n";
$values .= "\$date";
$content .= "if (\Error != 0) {\r\n";
$content .= "// import the style sheets for the forms (text is the defaults of the site)\r\n";
$content .= "echo \"<link rel=\\\"stylesheet\\\" type=\\\"text/css\\\"
href=\\\"$directory/style-main.css\\\">\";\r\n";
$content .= "echo \"<link rel=\\\"stylesheet\\\" type=\\\"text/css\\\"
href=\\\"$directory/style-forms.css\\\">\";\r\n";
$content .= "echo \"<p class=\\\"center\\\">\";\r\n";
$content .= "echo \msg;\r\n";
$content .= "echo \"<br />\";\r\n";
$content .= "echo \"<a href=\\\"Javascript:history.go(-1)\\\">Go Back</a>
and complete the form.\";\r\n";
$content .= "exit;\r\n";
$content .= "} else {\r\n";
$content .= "header('refresh:0; url='http://$domain');\r\n";
$content .= "\r\n";
$content .= "// produce the style sheets to format the text\r\n";
$content .= "echo \"<link rel=\\\"stylesheet\\\" type=\\\"text/css\\\"
href=\\\"$directory/style-main.css\\\">\";\r\n";
$content .= "echo \"<link rel=\\\"stylesheet\\\" type=\\\"text/css\\\"
href=\\\"$directory/style-forms.css\\\">\";\r\n";
$content .= "// connect to the database to call the operation\r\n";
$content .= "\$connection = @mysql_connect(\$sqlserver, \$dbuser, \$dbpass)
or die(mysql_error());\r\n";
$content .= "\$db = @mysql_select_db(\$dbname, \$connection) or die(mysql_error());\r\n";
$content .= "\r\n";
$content .= "// insert the values into the SQL table\r\n";
$content .= "\$sql = \"INSERT INTO $tablename VALUES ($values)\";\r\n";

```

```

$contents .= "\$result = @mysql_query(\$sql,\$connection) or die(mysql_error());\r\n";
$contents .= "\r\n";
$contents .= "// close the database connection\r\n";
$contents .= "mysql_close(\$connection);\r\n";
$contents .= "}\r\n";
$contents .= "?>\r\n";

return $contents;
}

// generate the table in the database
function valuesFormTable($idnum, $form_name, $appname) {
    $fileBackup = "../backup/$idnum.txt";
    $fileBackupHandle = @fopen($fileBackup, "r") or exit("Unable to open file!");

    $values = "(";
    while(!feof($fileBackupHandle))
    {
        $line = @fgets($fileBackupHandle);
        $field_array = explode("|", $line);

        if ($line == "") { break; }
        @extract(getPerLine($field_array));

        if ($datatype == "input") {
            $values .= "$field_name VARCHAR($input_length), ";
        } else if ($datatype == "textarea") {
            $values .= "$field_name VARCHAR(500), ";
        } else if ($datatype == "radio") {
            $values .= "$field_name VARCHAR(150), ";
        } else if ($datatype == "checkbox") {
            $values .= "$field_name VARCHAR(500), ";
        } else if ($datatype == "dropdown") {
            $values .= "$field_name VARCHAR(150), ";
        } else if ($datatype == "date") {
            $values .= "$field_name DATE, ";
        } else if ($datatype == "phone") {
            $values .= "$field_name VARCHAR(15), ";
        } else if ($datatype == "address") {
            $values .= "$field_name" . "address VARCHAR(250), ";
            $values .= "$field_name" . "city VARCHAR(150), ";
            $values .= "$field_name" . "state VARCHAR(2), ";
            $values .= "$field_name" . "zipcode VARCHAR(5), ";
        }
    }
    $values .= "date_added DATE)";

    return $values;
}

?>

```

/calendar/admin/action-add.php

```
<?php
// prevent caching of the web page
header("Expires: Sat, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: post-check=0, pre-check=0",false);
session_cache_limiter();

session_start();

// require the configuration files
require('../users-config.php');

// check to see if the user is logged in
if (!$_SESSION['user_name']) {
    $_SESSION['temp_redirect'] = "http://$domain" . $_SERVER['PHP_SELF'];
    header("Location:$users_directory/acct-login.html");
    exit;
}

// check to see if user is allowder access
$userrights = $_SESSION['userrights'];
if ($userrights != "Administrator") {
    header("Location:$users_directory/acct-noaccess.html");
    exit;
}

$errors = 0;
// determine form errors
if ($_POST['title'] == "" || $_POST['location'] == "" || $_POST['contact_name'] == "" ||
$_POST['contact_email'] == "") {
    // produce the style sheets to format the text
    echo "<link rel='stylesheet' type='text/css' href='../style-main.css'>";
    echo "<link rel='stylesheet' type='text/css' href='../style-forms.css'>";
    echo "<p align='center'>";
    echo "<div id='content'><p class='center red'>You must complete all of the fields, please <a
href='javascript:history.go(-1)'>Go Back </a>and complete all of the fields.<br />&nbsp;  <br />";

    // print the error
    if ($_POST['title'] == "") {
        echo "The title field was left empty<br />"; $errors++; }
    if ($_POST['location'] == "") {
        echo "The location field was left empty<br />"; $errors++; }
    if ($_POST['contact_name'] == "") {
        echo "The contact name field was left empty<br />"; $errors++; }
    if ($_POST['contact_email'] == "") {
        echo "The contact e-mail address field was left empty<br />"; $errors++; }

    echo "&nbsp;  <br />There were $errors in your submission.</p>";
}
```

```

echo "</div>";
exit;
} else {
// establish the page redirection for no errors
header("refresh:5; url='admin-ctlpanel.php'");

// produce the style sheets to format the text
echo "<link rel='stylesheet' type='text/css' href='../style-main.css'>";
echo "<link rel='stylesheet' type='text/css' href='../style-forms.css'>";
echo "<p class='center'>";

// STEP 1
$title = str_replace("","",$_POST['title']);
$occurance = $_POST['occurance'];
$location = str_replace("","",$_POST['location']);
$rainloc = $_POST['rainloc'];
if ($rainloc == "") { $rainloc = "No Information Provided"; }
$description = str_replace("","",$_POST['description']);
if ($description == "") { $description = "No Information Provided"; }

// STEP 2
$speaker = str_replace("","",$_POST['speaker']);
if ($speaker == "") { $speaker = "No Information Provided"; }
$webpage = $_POST['webpage'];
if ($webpage == "") { $webpage = "No Information Provided"; }
$sponsor = $_POST['sponsor'];
$category = $_POST['category'];

// STEP 3
$audience = $_POST['audience'];
if ($audience == "other") { $audience = str_replace("","",$_POST['audience_other']); }
$cost = str_replace("$","",$_POST['cost']);
if ($cost == "") { $cost = "0.00"; }
$cost_until = $_POST['cost_until'];
if ($cost_until == "YES") {
    $until_hour = (int)$_POST['until_hour'];
    $until_minutes = (int)$_POST['until_minutes'];
    $until_meridiem = $_POST['until_meridiem'];
    if ($until_meridiem == "p.m.") { $until_hour = $until_hour + 12; }
}
$tickets = $_POST['tickets'];
if ($tickets == "") { $tickets = "No Information Provided"; }
$contact_name = str_replace("","",$_POST['contact_name']);
$contact_phone = $_POST['contact_phone'];
if ($contact_phone == "") { $contact_phone = "No Information Provided"; }
$contact_email = $_POST['contact_email'];

// STEP 4
$idnum = $_POST['idnum'];
$post = $_POST['postevent'];

```



```

// Execute the functions based on the Event Date, Time, and Occurances
// aquire the start date
$startdate_month      = (int)$_POST['startdate_month'];
$startdate_day        = (int)$_POST['startdate_day'];
$startdate_year       = (int)$_POST['startdate_year'];
$starttime_hours      = (int)$_POST['starttime_hours'];
$starttime_minutes    = (int)$_POST['starttime_minutes'];
$starttime_meridiem   = $_POST['starttime_meridiem'];

// convert the hours into a 24 hour timestamp
if ($starttime_meridiem == "p.m.") { $starttime_hours = $starttime_hours + 12; }

// produce the UNIX timestamp and return the proper date format
$start_unix = mktime($starttime_hours, $starttime_minutes, 0, $startdate_month, $startdate_day,
$startdate_year);

// aquire the end date
$duration_hours      = (int)$_POST['duration_hours'];
$duration_minutes    = (int)$_POST['duration_minutes'];
$duration            = ($duration_hours * 3600) + ($duration_minutes * 60);
$dur_unix = mktime($duration_hours, $duration_minutes, 0, $startdate_month, $startdate_day,
$startdate_year);

// produce the UNIX timestamp and return the proper date format
$end_unix = $start_unix + $duration;

// determine the cost relation in time
$cost_until = $_POST['cost_until'];
if ($cost_until == "YES") {
    $cost_unix = mktime($until_hour, $until_minutes, 0, $startdate_month, $startdate_day,
$startdate_year);
} else {
    $cost_unix = "0";
}

// produce a date output based on the occurances
$enddate_month      = (int)$_POST['enddate_month'];
$enddate_day        = (int)$_POST['enddate_day'];
$enddate_year       = (int)$_POST['enddate_year'];
$endtime_hours      = $starttime_hours;
$endtime_minutes    = $starttime_minutes;
$day_stop           = mktime($endtime_hours, $endtime_minutes, 0, $enddate_month,
$enddate_day, $enddate_year);

// connect to the database to call the operation
$connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
$db = @mysql_select_db($dbname,$connection)          or die(mysql_error());

// query the database to call the operation
$sql = "SELECT * FROM $tblname WHERE idnum = '$_POST[idnum]'";
$result = @mysql_query($sql,$connection) or die(mysql_error());

```

```

// aquire the number of matching rows
$rownum = mysql_num_rows($result);

// assign a new event id num
if ($rownum != 0) {
    $idnum = rand(1000,1999) . rand(1000000,9999999);
    echo "<span style=\"color:red;font-size:11px;\">The Event ID# has been changed.
        Please note the new assignment.</span><br />";
} else {
    echo "<span style=\"color:green;font-size:11px;\">The Event has been assigned the
        ID# $idnum.</span><br />";
}

// process the SQL request based on the occurrence model
if ($occurrence == "O") {
    // insert the values into the SQL table
    $sql = "INSERT INTO $tblname VALUES
    ('$title', '$start_unix', '$end_unix', '$dur_unix', '$location', '$rainloc', '$description',
    '$speaker', '$webpage', '$category', '$sponsor', '$audience', '$cost', '$cost_unix', '$tickets',
    '$contact_name', '$contact_phone', '$contact_email', '$post', '$idnum)";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
    echo "<span style=\"color:green;font-size:11px;\">Adding new record...</span><br />";
    // close the database connection
    mysql_close($connection);
    echo "<span style=\"color:green;font-size:11px;\">
        Your event was successfully created.<br />";
    echo "You will be redirected to calendar management page in five seconds.</span>";
} else if ($occurrence == "D") {
    while ($start_unix <= $day_stop)
    {
        // insert the values into the SQL table
        $sql = "INSERT INTO $tblname VALUES
        ('$title', '$start_unix', '$end_unix', '$dur_unix', '$location', '$rainloc',
        '$description', '$speaker', '$webpage', '$category', '$sponsor', '$audience', '$cost',
        '$cost_unix', '$tickets', '$contact_name', '$contact_phone', '$contact_email',
        '$post', '$idnum)";
        $result = @mysql_query($sql,$connection) or die(mysql_error());
        echo "<span style=\"color:green;font-size:11px;\">
            Adding new record...</span><br />";
        $start_unix = strtotime("+1 day", $start_unix);
        $end_unix = strtotime("+1 day", $end_unix);
        $idnum = rand(1000,1999) . rand(1000000,9999999);
        echo "&nbsp;<br /><span style=\"color:green;font-size:11px;\">
            A new Event ID# has been assigned: $idnum.</span><br />";
    }
    // close the database connection
    mysql_close($connection);
    echo "<span style=\"color:green;font-size:11px;\">
        Your event was successfully created.<br />";
    echo "You will be redirected to calendar management page in five seconds.</span>";
}

```

```

} else if ($occurance == "W") {
    while ($start_unix <= $day_stop)
    {
        // insert the values into the SQL table
        $sql = "INSERT INTO $tblname VALUES
('$title', '$start_unix', '$end_unix', '$dur_unix', '$location', '$rainloc',
'$description', '$speaker', '$webpage', '$category', '$sponsor', '$audience', '$cost',
'$cost_unix', '$tickets', '$contact_name', '$contact_phone', '$contact_email',
'$post', '$idnum)";
        $result = @mysql_query($sql,$connection) or die(mysql_error());
        echo "<span style=\"color:green;font-size:11px;\">
            Adding new record...</span><br />";
        $start_unix    = strtotime("+1 week", $start_unix);
        $end_unix      = strtotime("+1 week", $end_unix);
        $idnum         = rand(1000,1999) . rand(1000000,9999999);
        echo "&nbsp;<br /><span style=\"color:green;font-size:11px;\">
            A new Event ID# has been assigned: $idnum.</span><br />";
    }
    // close the database connection
    mysql_close($connection);
    echo "<span style=\"color:green;font-size:11px;\">
        Your event was successfully created.<br />";
    echo "You will be redirected to calendar management page in five seconds.</span>";
} else if ($occurance == "Y") {
    while ($start_unix <= $day_stop)
    {
        // insert the values into the SQL table
        $sql = "INSERT INTO $tblname VALUES
('$title', '$start_unix', '$end_unix', '$dur_unix', '$location', '$rainloc',
'$description', '$speaker', '$webpage', '$category', '$sponsor', '$audience', '$cost',
'$cost_unix', '$tickets', '$contact_name', '$contact_phone', '$contact_email',
'$post', '$idnum)";
        $result = @mysql_query($sql,$connection) or die(mysql_error());
        echo "<span style=\"color:green;font-size:11px;\">
            Adding new record...</span><br />";
        $start_unix    = strtotime("+1 year", $start_unix);
        $end_unix      = strtotime("+1 year", $end_unix);
        $idnum         = rand(1000,1999) . rand(1000000,9999999);
        echo "&nbsp;<br /><span style=\"color:green;font-size:11px;\">
            A new Event ID# has been assigned: $idnum.</span><br />";
    }
    // close the database connection
    mysql_close($connection);
    echo "<span style=\"color:green;font-size:11px;\">
        Your event was successfully created.<br />";
    echo "You will be redirected to calendar management page in five seconds.</span>";
}
}
}
?>

```

/massmail/admin/action-email.php

```
<?php
// prevent caching of the web page
header("Expires: Sat, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: post-check=0, pre-check=0",false);
session_cache_limiter();

session_start();

// require the configuration and the functions files
require('../users-config.php');

// check to see if the user is logged in
if (!$_SESSION['user_name']) {
    $_SESSION['temp_redirect'] = "http://$domain" . $_SERVER['PHP_SELF'];
    header("Location:$directory/acct-login.html");
    exit;
}

// check to see if user is allowder access
$userrights = $_SESSION['userrights'];
if ($userrights != "Administrator") {
    header("Location:$directory/acct-noaccess.html");
    exit;
}

// define the variables
$from      = $_POST['name'] . " <" . $_POST['reply'] . ">";
$cc        = $_POST['ccopy'];
$bcc       = $_POST['bccopy'];
$userrights = $_POST['userrights'];
$format    = $_POST['format'];
$subject   = $_POST['subject'];
$message   = $_POST['message'];
$personal  = $_POST['personal'];

if ($subject != "") {
    if ($message != "") {
        // establish the page redirection for no errors
        header("refresh:15; url='admin-ctlpanel.php'");

        // call the formatting stylesheets
        echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
        echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";

        // connect to the database to call the operation
        $connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
        $db = @mysql_select_db($dbname,$connection) or die(mysql_error());
```

```

// query the database to call the operation
if ($userrights == "All User Groups") {
    $sql = "SELECT * FROM $tblname";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
} else {
    $sql = "SELECT * FROM $tblname WHERE userrights = '$userrights'";
    $result = @mysql_query($sql,$connection) or die(mysql_error());
}

while ($sql = mysql_fetch_object($result))
{
    $first = $sql -> firstname;
    $last = $sql -> lastname;
    $saddy = $sql -> email;

    if ($format == "HTML") {
        // Always set content-type when sending HTML email
        $headers = "MIME-Version: 1.0" . "\r\n";
        $headers .= "Content-type:text/html;charset=iso-8859-1" . "\r\n";
        $headers .= "From: $from" . "\r\n";
        $headers .= "Reply-To: $from" . "\r\n";
        if ($cc != "") { $headers .= "Cc: $cc" . "\r\n"; }
        if ($bcc != "") { $headers .= "Bcc: $bcc" . "\r\n"; }
    } else if ($format == "PLAIN") {
        $headers = "From: $from" . "\r\n";
        $headers .= "Reply-To: $from" . "\r\n";
        if ($cc != "") { $headers .= "Cc: $cc" . "\r\n"; }
        if ($bcc != "") { $headers .= "Bcc: $bcc" . "\r\n"; }
    }

    if ($personal == "YES" && $format == "HTML") {
        $msg = "Hello $first $last,<br /><br />";
        $msg .= $message;
    } else if ($personal == "YES" && $format == "PLAIN") {
        $msg = "Hello $first $last,\r\n";
        $msg .= "\r\n";
        $msg .= $message;
    } else {
        $msg = $message;
    }

    mail($saddy, $subject, $msg, $headers);
}
} else {
    echo "<hr>";
    echo "<p class='center'>The message field must not be left blank.</p>";
}
} else {
    echo "<hr>";
    echo "<p class='center'>The subject field must not be left blank.</p>";
}

```

```

        exit;
    }
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Account Management | Add User</title>
    <link rel="stylesheet" type="text/css" href="style-forms.css" />
    <link rel="stylesheet" type="text/css" href="style-main.css" />
</head>

<body>
<p>&nbsp;&nbsp;&nbsp;</p>

<?php
echo "<p class='center'>This page will refresh in fifteen seconds.<br />The following E-mail has been
sent:</p>";
echo "<br />";

echo "<div id='userslet'>";

echo "<strong>Sender</strong>: ";
echo $_POST['name'] . " &lt;" . $_POST['reply'] . "&gt;" . "<br />";

echo "<strong>User Groups</strong>: ";
echo "$userrights<br />";

echo "<strong>Subject</strong>: ";
echo "$subject<br />";

echo "<strong>Message</strong>:<br /> ";
echo "$message";
echo "<br />";

echo "</div>";

echo "<br clear='all'>";
?>
</body>
</html>

```

/users/admin/action-add.php

```
<?php
// prevent caching of the web page
header("Expires: Sat, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: post-check=0, pre-check=0",false);
session_cache_limiter();

session_start();

// call the formatting stylesheets
echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-main.css\">";
echo "<link rel=\"stylesheet\" type=\"text/css\" href=\"../style-forms.css\">";

// require the configuration and the functions files
require('./users-config.php');
require('./users-functn.php');

// check to see if the user is logged in
if (!$_SESSION['user_name']) {
    $_SESSION['temp_redirect'] = "http://$domain" . $_SERVER['PHP_SELF'];
    header("Location:$directory/acct-login.html");
    exit;
}

// check to see if user is allow access
$userrights = $_SESSION['userrights'];
if ($userrights != "Administrator") {
    header("Location:$directory/acct-noaccess.html");
    exit;
}

//checks password length
if (isset($_POST['action_add'])) {

    if (AcceptPassword($minlength, $maxlength, $_POST['password']) == "no") {
        ?>
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
        <html xmlns="http://www.w3.org/1999/xhtml">

        <head>
            <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
            <title>Account Management | Change Password</title>
            <link rel="stylesheet" type="text/css" href="style-forms.css" />
            <link rel="stylesheet" type="text/css" href="style-main.css" />
            <script language="JavaScript">
            <!--
            function ChangePass(msg) {/v1.0
            alert(msg);
```

```

    }
    // -->
</script>
</head>

<body onLoad="ChangePass('The password has to be between
    <? echo $minlength; ?> & <? echo $maxlength; ?> characters.')">

</body>
</html>
<?
exit;
}

}

if (isset($_POST['action_add'])) {
    // define variables

    // check requirement to change password
    $reqchange = $_POST['reqchange'];
    if ($reqchange == "YES") {
        $pchange = "1";
    } else {
        $pchange = "0";
    }

    // check requirment to verify email
    $sendmail = $_POST['sendmail'];
    if ($sendmail == "YES") {
        $emailver = "NO";
    } else {
        $emailver = "YES";
    }

    // connect to the database to call the operation
    $connection = @mysql_connect($sqlserver, $dbuser, $dbpass) or die(mysql_error());
    $db = @mysql_select_db($dbname,$connection) or die(mysql_error());

    // query the database to call the operation
    $sql = "SELECT * FROM $tblname WHERE username = '$_POST[username]'";
    $result = @mysql_query($sql,$connection) or die(mysql_error());

    // aquire the number of matching rows
    $rownum = mysql_num_rows($result);

    // test match the username and password from the database
    if ($rownum != 0) {
        echo "<p class='center'>Sorry, that username already exists.<br />";
        echo "Try Another Username.</p>";
        exit;
    }
}

```



```

    } else {
        $sql = "INSERT INTO $tblname VALUES
        ('$_POST[firstname]', '$_POST[lastname]', '$_POST[username]',
        password('$_POST[password]'), 'User', '$_POST[changerights]', '$pchange',
        '$_POST[email]', '$redirect', '$mailver', ")";
        $result = @mysql_query($sql,$connection) or die(mysql_error());
    }

    if ($sendmail == "YES") {
        $mailheaders = "From: $domain\n";
        $mailheaders .= "Dear $_POST[firstname] $_POST[lastname],\n";
        $mailheaders .= "\n";
        $mailheaders .= "An account has been created for you at $domain.\n";
        $mailheaders .= "You have been assigned the following credentials:\n";
        $mailheaders .= "Username: $_POST[username]\n";
        $mailheaders .= "Password: $_POST[password]\n";
        $mailheaders .= "\n";
        $mailheaders .= "Please login at:\n";
        $mailheaders .= "$directory/acct-login.html\n";
        $mailheaders .= "\n";
        $mailheaders .= "Should you have any complications,
            please email the System Administrator at:\n";
        $mailheaders .= "$email_admin\n";
        $mailheaders .= "\n";
        $mailheaders .= "\n";
        $mailheaders .= "-----\n";
        $mailheaders .= "This is an automated message from $domain\n";
        $mailheaders .= "If you believe you are receiving this message in error,\n";
        $mailheaders .= "please contact the administrator at $email_admin.\n";

        $to = "$_POST[email]";
        $subject = "An Account Has Been Created for You at $domain";

        mail($to, $subject, $mailheaders, "From: $domain <$email_admin>\n");
    }
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Account Management | Add User</title>
    <link rel="stylesheet" type="text/css" href="style-forms.css" />
    <link rel="stylesheet" type="text/css" href="style-main.css" />
</head>

<body>

<?php

```

```

if (!isset($_POST['action_add'])) {
    echo "<hr>";
    echo "<p class='center'>No action has been taken yet.</p>";
} else {
    echo "<hr>";
    echo "<p class='center'>The following user was created.</p>";
    echo "<br />";

    echo "<div id='userslct'>";

    echo "<div id='username'>";
    echo "<p><strong>$_POST[username]</strong></p>$_POST[changerights]";
    echo "</div>";

    echo "<div id='userdesc'>";
    echo "<p>Name:<br />E-Mail:<br />Password Change:
        <br />E-Mail Verified:<br />Usergroup:<br />Default Redirect:</p>";
    echo "</div>";

    echo "<div id='uservalue'>";
    echo "<p>$_POST[firstname] $_POST[lastname]<br />
        <a href='mailto:$_POST[email]'>$_POST[email]</a>
        <br />$_POST[reqchange]<br />$_POST[mailver]<br />User<br />$_POST[redirect]</p>";
    echo "</div>";

    echo "</div>";

    echo "<br clear='all'>";
}
?>
</body>
</html>

```