A Comparison Study of Web Based Application Development Using PHP and ASP.NET

Morris M. Liaw, Ph. D.
Univ. of Houston Clear Lake
Houston, TX 77058, US
liaw@uhcl.edu

Abdul Mansoor Mohammed.
Univ. of Houston Clear Lake
Houston, TX 77058, US
inform.mansoor@gmail.com

Abstract:

This paper takes a detailed look at PHP, an open source server-side scripting language, and compares it to its heavily marketed counterpart ASP.NET. It explores various concepts which are pertinent to choosing programming platforms and outlines the advantages of each language over the other. A sample application called "Personal Media Center" is developed in PHP to illustrate these concepts and advantages, and a summary of the factors which affect the usability of both platforms is discussed at the conclusion of this paper.

Keywords: PHP, MySQL, Web Application Platform, ASP.NET

Acknowledgement:

Very special thanks to Matthew E. Liaw (gordongartrell@gmail.com), Stanford University for proof reading the content of this paper.

1. Introduction:

When developing a web application that has server-side scripting capabilities, the natural choice for a programmer today would be one of the two most widely used scripting languages, namely Microsoft`s ASP.NET and/or open source PHP. Though the developers of ASP.NET and PHP assure their users that both platforms are capable of fulfilling their web application needs, ASP.NET requires the support of a much larger development community. Why this is the case is worth investigating since both languages have been utilized in creating very large and very successful web applications.

2. Brief Review of PHP:

As mentioned earlier, PHP is primarily focused on server-side scripting capabilities to provide programmers with the necessary tools for creating web applications that are heavily dynamic in nature. It is capable of performing all the actions of any CGI program and has a vast array of functionalities, e.g. collecting form data, generating dynamic web pages, and setting and accessing cookies for session control. The primary functionalities of PHP, as stated by its developers, are server-side scripting, command line scripting, and desktop applications. It can be used on all major operating systems, including UNIX and Linux, and thus, gives web developers a choice of operating system and web server when deploying an application. Another strong and significant feature of PHP is its ability to support various databases. A database enabled web page can be written with incredible simplicity using either database specific extensions for MySQL or abstraction layers like PDO [PHP Database Objects].

Based on a 2007 Netcraft survey, 54.9% [112,945,968] of 205,714,253 websites were hosted on Apache HTTP Servers whereas 25.9% [53,217,620] were hosted on Windows Servers. This information appears to indicate that PHP is a far more popular platform than ASP.NET for web-application developers. Figure 1 shows the statistics for this survey.[1]

Since we will be looking at PHP from the perspective of an entry level programmer, we will only establish and verify its basic functionalities. We do this by developing a simple 3-tier website called Personal Media Center.

The following section discusses the application in more detail.

3. **About Personal Media Center:**

The proposed functionalities of Personal Media Center are those of a website designed primarily to host different types of media such as pictures, music, and videos which can be remotely uploaded by the website's administrator. Subscribers and visitors from all over the world will able to view this media, and registered users will be able to discuss and post comments about the media in a manner akin to other social networking websites. Users will also be able to register for updates and newsletters sent out by the administrator.

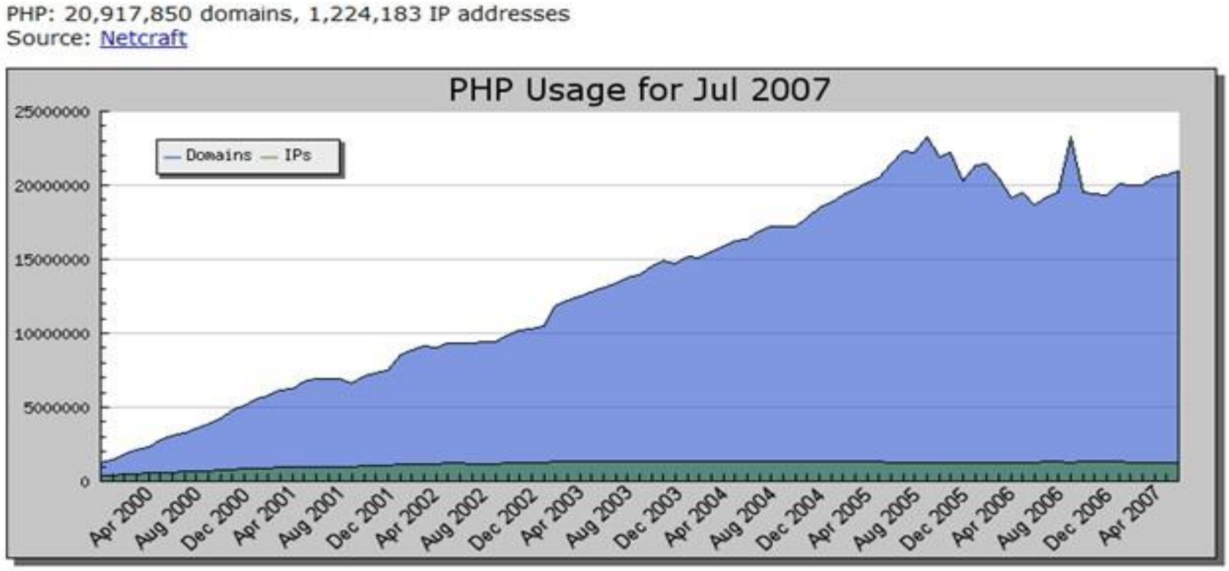


Figure 1 – Netcraft survey results for PHP Usage through July 2007.

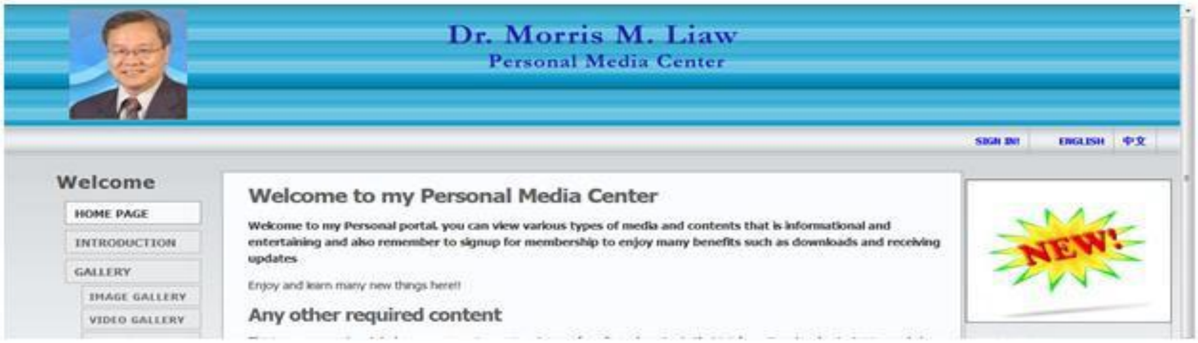
The performance of PHP is put to the test by using it to dynamically manipulate the entire webpage`s text into either Simplified Chinese or English based on the user`s choice of language.

The application is designed with PHP technology to run on a web server running Apache Tomcat 6.0 and extensively uses MySQL 5.5 for all database-related operations. Although any viewer with an internet connection has the ability to browse content on the website, only the administrator has the right to upload media to it. Additionally, only registered users will be able to comment on the media.

The application uses the cookie functions in PHP API to execute these authentication mechanisms. There is a single interface for all users, but the rights are based on the user’s login credentials. This fully functional application is deployed on the DCM (Division of Computing and Mathematics) server at the University of Houston-Clear Lake and can be tested at the following link.

http://sceweb.sce.uhcl.edu/liaw/mediacenterv2.0/

The following figures show screenshots of pages on the Personal Media Center application website. Figure 2(a) shows the home page of the application, and Figure 2(b) shows the gallery page of the website which serves as the root page for browsing all media.



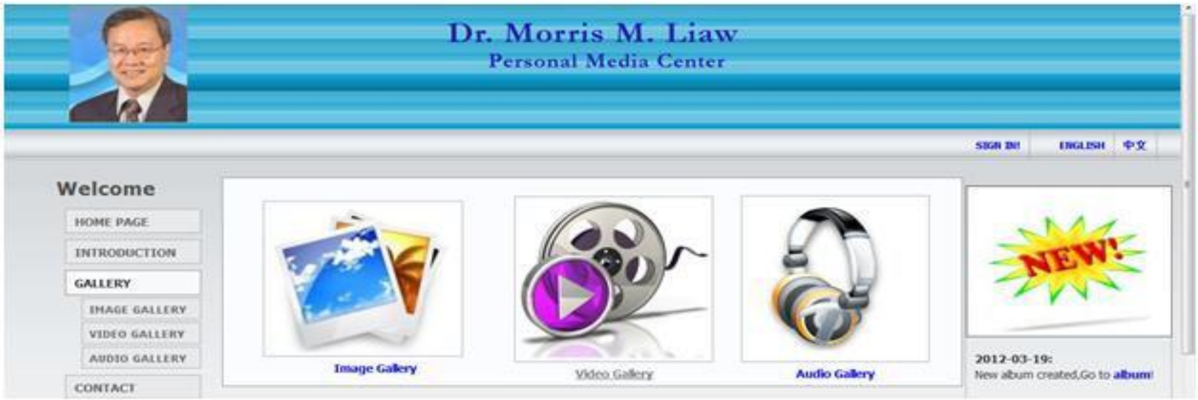
Welcome to my Personal Media Center

Welcome to my Personal portal. you can view various types of media and contents that is informational and entertaining and also remember to signup for membership to enjoy many benefits such as downloads and receiving updates

Enjoy and learn many new things here!!

Any other required content

Figure 2(a) - Home page for the application.



2012-03-19:
New album created,Go to album!

Figure 2(b) - Gallery page for the application.

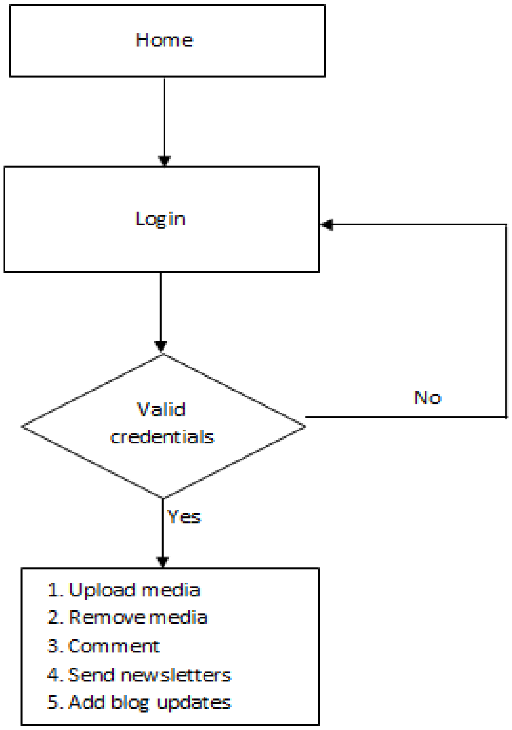
For the Personal Media Center application, there are three types of users as mentioned below. Each class of user has a different level of access, with the administrator having the highest level and casual users having the lowest level of access control rights.

1. Administrator
2. Registered users
3. Casual viewers

Figure 3 below is a simple flow chart of actions for the administrator of the application. Some of the additional functions not mentioned in the above chart can be found in the application but have been omitted for the sake of brevity.

4. Using PHP for Applications:

One of the biggest advantages of using PHP when creating web applications is that PHP scripts can be embedded directly into HTML pages. During the initial development phase of the Personal Media Center application, a static HTML website was designed using common HTML and various text editors. PHP scripts and actions were embedded into the HTML pages later on which helped to simplify the development process by separating interface development from functional implementation.



1. Upload media:	Upload media to one of the declared categories.
2. Remove media:	Remove any previously uploaded media.
3. Comment:	Comment or respond to other user comments on media.
4. Send newsletters:	Send monthly newsletters to subscribed users.
5. Add blog updates:	Add updates to the blog feed.

Figure 3 – Available administrator actions in Personal Media Center.

The major components of functionality from an administrator's perspective are uploading and managing media. The following snippets of code in Figures 4(a) and 4(b) show the PHP functions used to upload and delete files, respectively. *move_uploaded_file* takes the path of the file to be uploaded and the future path of the file at the server as arguments and moves the file to the server, thus completing the upload. *unlink* [2] takes the path of the file and simply deletes it from the server.

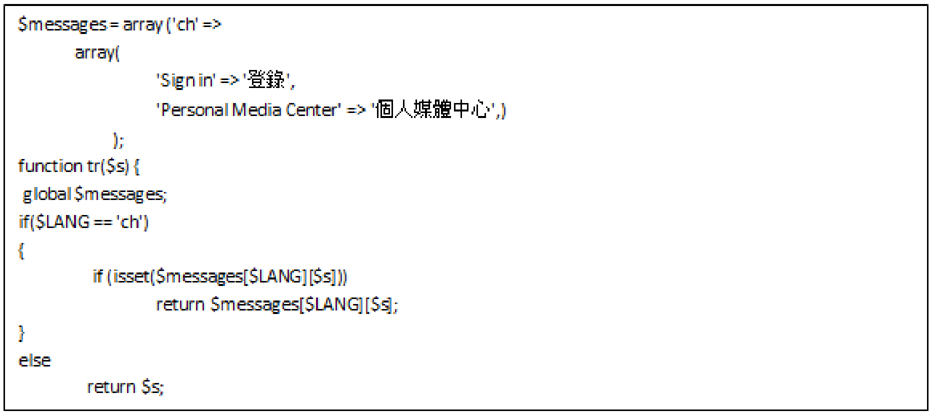
As mentioned previously, a translation function built into the application can be used to test the performance of PHP. The function is able to translate the text within Personal Media Center by using a static dictionary of user-defined terminology.

The translation function is called for each and every text phrase in all of the pages of the website.

Each function call follows the sequence of actions below:

1. Check the user-set cookie for language preference; if none is found, set the cookie to English as the default preference.
2. Look up the phrase in the dictionary of defined phrases.
3. Return the phrase in the corresponding language found in the cookie.

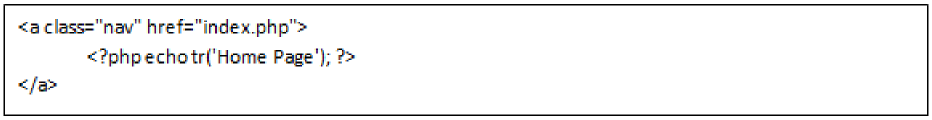
Currently, this may not be the best implementation of a translation function as one must specify each and every phrase in the dictionary. However, the translation function plays a key role in testing PHP's ability to handle large numbers of function calls from a specific client request. A snippet from the translation dictionary is shown in Figure 5(a).



```
$messages = array ('ch' =>
        array(
                    'Sign in' => '登錄',
                    'Personal Media Center' => '個人媒體中心',)
             );
function tr($s) {
 global $messages;
if($LANG == 'ch')
{
          if (isset($messages[$LANG][$s]))
                    return $messages[$LANG][$s];
}
else
         return $s;
```

Figure 5(a) - Sample dictionary translation function code.

Translating a statement is done by calling the custom-defined '*tr*' function as shown in Figure 5(b).



```
<a class="nav" href="index.php">
         <?php echo tr('Home Page'); ?>
</a>
```

Figure 5(b) - *tr* function call.

5. Using MySQL:

All database-related back-end operations for Personal Media Center were done using MySQL. An important reason for choosing MySQL as a database back-end was PHP's built-in support for it. The MySQL query generator in PHP is very useful in the formulation and testing of queries.

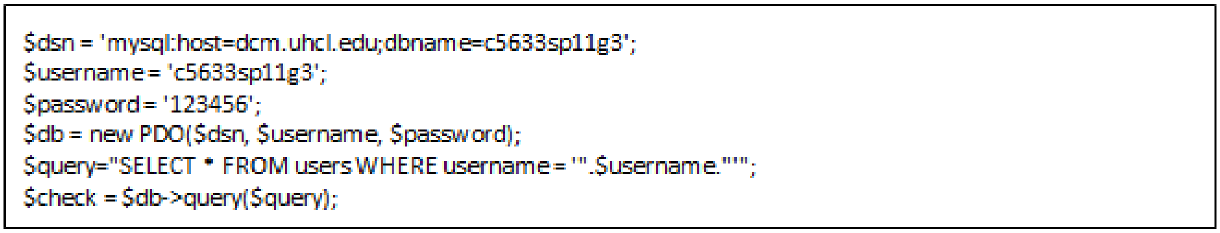
The functions *open* and *close* are not required for opening and closing connections to MySQL servers. A connection initialization, query declaration, and execution request is shown in the snippet of code in Figure 6(a).

The result set of a select query is stored in the requesting variable and can be accessed as shown in Figure 6(b).

6. **Conclusion:**

The development of Personal Media Center shows that PHP has several advantages over Microsoft's ASP.NET. The following crucial factors were analyzed to reach this conclusion-

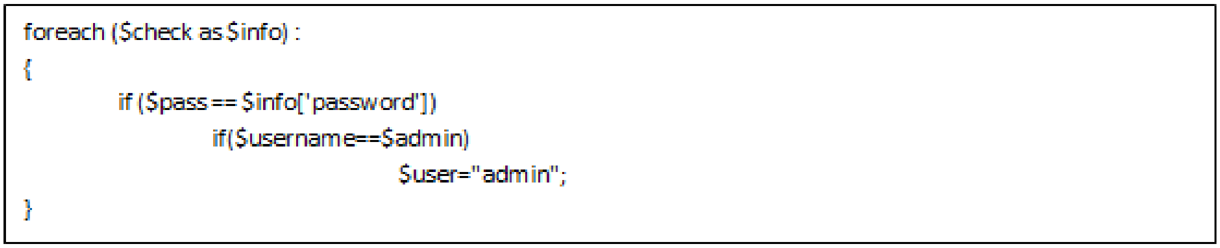
(i) Cost: When compared to ASP.NET, PHP leads in this area because it is free, and no additional licensing fee is required to use it. Each member of the PHP suite, known collectively as LAMP (Linux, Apache, MySQL, and PHP), is also free, with support and upgrades provided at no extra cost. On the other hand, licensing Microsoft's server software can cost over $1000. This makes LAMP a very popular platform for designing inexpensive yet effective applications.[3]



```
$dsn = 'mysql:host=dcm.uhcl.edu;dbname=c5633sp11g3';
$username = 'c5633sp11g3';
$password = '123456';
$db = new PDO($dsn, $username, $password);
$query="SELECT * FROM users WHERE username = '".$username."'";
$check = $db->query($query);
```

Figure 6(a) – Sample code for a connection initialization, query declaration, and execution request.

The result set of a select query is stored in the requesting variable and can be accessed as shown in Figure 6(b).



```
foreach ($check as $info) :
{
        if ($pass == $info['password'])
                if($username==$admin)
                                $user="admin";
}
```

Figure 6(b) – Accessing the result set of a select query.

(ii) Performance: When the performance of the programming language is considered alone irrespective of database engine used, PHP has an advantage over ASP.NET because it is known for running “lightly” on servers and is free of any unneeded packages. The lack of GUI’s also gives PHP servers an extra edge in performance.

(iii)Complexity: Tasks can be accomplished with far fewer lines of code in PHP than in ASP.NET. Though coding outside of HTML pages may initially appear to simplify the development process, having to map related code between separate files when using ASP.NET requires a high level of understanding which can act as an obstacle to more novice programmers. It is no mystery then why PHP has become very popular with beginners.

Future studies will explore why the .NET framework is still popular with many developers, despite several advantages associated with PHP and the LAMP software suite.

7. References:

[1] Web Resource: www.php.net

[2] Murach, Joel, and Ray Harris. Murach's PHP and MySQL: Training & Reference. [Fresno, Calif.]: Mike Murach & Associates, 2010. Print.

[3] Kohan, Bernard. "PHP vs ASP.net Comparison." PHP vs ASP.net Performance Comparison. Bernard Kohan. Web. 20 Mar. 2012. <http://www.comentum.com/php-vs-asp.net-comparison.html>.

[4] Web Resource For Zend: [http://www.zend.com/en/]