An Empirical Comparison Of: HTML, PHP, COLDFUSION, PERL, ASP.NET, JAVASCRIPT, VBSCRIPT, PYTHON AND JSP

Amadin I.F.¹, Nwelih E.²

Abstract- With the advent of the World Wide Web, several Web development languages have emerged and selecting a suitable one is never an easy task. Over the years, several attempts have been made to evaluate Web development tools vis-à-vis software measurement. This paper presents an experimental evaluation of nine web development languages. A shopping cart application was implemented in each of the Web development languages and the following factors were used in our evaluation: Platform, Performance, Functionality, Ease of use, Reliability, Program length, Portability, Database supports, Speed of execution, Maintainability, Object oriented programming, and Development cost.

Keywords- Web Development languages, World Wide Web, Information Technology (IT) and Web browser

I. INTRODUCTION

The World Wide Web (WWW) has rapidly become the standard for displaying information on the Internet. Over the last ten years, the nature of the Internet and the World Wide Web has changed drastically from what they were then. Exciting new developments occur almost daily, as the pace of innovation is unprecedented by any other technology (Berner-Lee, 2002). This tremendous advancement is mainly due to improvements in the design of Web sites with the aid of sophisticated Web development technologies and languages. The advent of these languages has changed the content of the Internet from its usual Web pages full of only static text and very few images to web sites that not only animate text and images but offer a wide range of services including database and multimedia features. Terms like portals, e-(commerce, payment, education, banking, learning etc) have become everyday terms due to the improvement in Web development technologies and their familiarity with the general public.

Several methods and experiments have been carried out to determine the suitability and effectiveness of Web development tools which have led to the emergence of three evaluation methods namely: Empirical, Vendor and Usability. The metrics for comparing Web development tools coding and design representations is reported in Ovum (2000). Work on assessing an aspect of a visual tools coding and design representations is reported in Ovum Usability. The metrics for comparing Web development evaluation methods namely: Empirical, Vendor and Development cost.

II. BACKGROUND

A programming language is simply referred to as a system of communication with its own set of conventions and special words used to interact with the computer system. A programming language enables a programmer to dictate what, how and when a computer system will perform a task. In this modern age of information technology (IT), where the computer and the internet has now become a key player in every area of our lives, the need for a comparative study on the different languages used to interact with the computer and internet has become necessary. According to Janstal (2000), Web development tools are failing to address users’ needs despite the promises made by vendors. However, the market has no clear leader, and there are inadequate products for medium or large-scale development projects, (Ovum, 1997). According to Mahar (1997), the main deficiencies in web development tools are that they cannot support teams of developers working together. In this fast evolving area, according to Ward-Dutton (2002) in his works, he analysed a range of tools for building integrated Web applications. Developers are continuously in search of tools or technologies available in the software market that suit their need, as thousands of Web development tools or technologies exist. Web development technologies are equally subject to the Laws of Evolution (survival of the

About¹- Department of Computer Science Faculty of Physical Sciences
University of Benin P.M.B 1154, Benin City
Nigeriafrankamadin@yahoo.com

About²- Department of Computer Science Faculty of Physical Sciences
University of Benin P.M.B 1154, Benin City
Nigeriaemmanuelnwelih@yahoo.com
fittest) and there are some criteria used in measuring their acceptability and usability. To measure the usability of Web development tools or technologies which are particularly useful for the development of dynamic Web pages and animated movies, Green and Petre (1996) introduced the popular Cognitive Dimensions framework which is a breadth-first evaluation technique for interactive devices and for non-interactive notations. It sets out a small vocabulary of terms designed to capture the cognitively-relevant aspects of structure, and shows how they can be traded off against each other. The development and application of various metrics for comparing visual and textual representations is reported in Nickerson (1994). Some common web development tools are discussed below:

**HTML**: Hyper Text Markup Language (HTML) which was founded in 1980 by Tim Berners-lee, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. (Tim, 2001)

**PHP**: Hypertext Preprocessor (PHP), was conceived in 1994 by Rasmus Lerdorf. He wrote it as a way to track visitors to his online CV. The first version was released in early 1995; Rasmus had found that by making the project open-source, people would fix his bugs. The first version was very straightforward and had a simple parser which recognized a few special macros and provided some of the utilities which were in common usage on web-pages back then. In 1995 it was renamed PHP/FI version 2. The "FI" in this version stood for the Form Interpreter which he added to PHP to cope with the growing needs of web-pages and mSQL (Microsoft SQL) support was also added at this time. PHP/FI underwent massive growth, and other people started to contribute code to it regularly. In 1997, Zeev Suraski and Andi Gutmans rewrote the main parser, and PHP shifted from being Rasmus' own to a more group orientated project. This formed the basis for PHP3, now named PHP: Hypertext Preprocessor with version such as PHP4 and PHP5 engine. The latest version comes with most of those features which were not in the earlier versions of PHP. It is easier to integrate into existing HTML pages, faster and more efficient for complex programming tasks and trying out new ideas. PHP is generally referred to as more stable and less resource intensive as well. (Valade, 2004)

**COLDFUSSION**: ColdFusion (CF) is an application server and software language used for Internet application development such as for dynamically-generated web sites. In this regard, ColdFusion is a similar product to Microsoft Active Server Pages, Java Server Pages or PHP. ColdFusion was the first amongst these technologies to provide the developer the capability of creating dynamic websites that were attached to a backend database i.e. Cold Fusion has better database abstraction. The primary distinguishing feature of ColdFusion is its associated scripting language, ColdFusion Markup Language (CFML), which compares to Active Server Pages, JSP, or PHP resembles HTML in syntax. "ColdFusion" is often used synonymously with "CFML", but there are additional CFML application servers besides ColdFusion, as ColdFusion supports programming languages other than CFML, such as server-side Actionscript and embedded scripts that can be written in a JavaScript-like language known as CFScrip. ColdFusion was originally developed by brothers JJ and Jeremy Allaire in July 1995. In 2001 Allaire was acquired by Macromedia, which in turn was acquired by Adobe Systems in 2005. ColdFusion is most often used for data-driven web sites or intranets, but can also be used to generate remote services such as SOAP web services or Flash remoting. It is especially well-suited as the server-side technology to the client-side Flex. ColdFusion can also handle asynchronous events such as SMS and instant messaging via its gateway interface; it also has a good error handling capability, date parsing features and more. ColdFusion is available on Win32, Solaris, Linux and HP/UX operating systems respectively. Published by Allaire in 2004, ColdFusion 4.0 is an enterprise level Web application development suite. This means it can be used not just to develop simple Web pages but also to develop databases, and more dynamic Web sites. The product is now in its fourth version and has been a consistent market leader. ColdFusion uses its own anatomized scripting tags, which when embedded in a Web page are read by the Web server. The Web server then produces dynamic output for the end user. All ColdFusion scripting is browser independent, making its content available to a wide audience. Only in the exception that CFML is combined with DHTML will a high end browser be needed. Instructions are passed to ColdFusion using templates. A template looks like any HTML file, and the only difference being the CFML tags. (http://www.adobe.com/products/coldfusion. Retrieved, 2010)

**PERL**: The first version was introduced in the year 1987 by Larry Wall. The author’s purpose for the creation was as a result of the disappointing result of languages such as sed, C, awk and the Bourne Shell offered him. He looked for a language that will combine all of their best features, while having a few disadvantages of its own. Since then, Perl has seen several versions of each additional function. Perl version 5, which was released in 1994, was a complete rewrite of the Perl interpreter, and introduced such things as hard references, modules, objects and lexical scoring. Several minor versions of Perl appeared since then, and the most up-to-date stable version (as of October 2005) is 5.8.x. Perl became especially popular as a language for writing server-side scripts for web-servers. But that's not the only use of perl, as it is commonly used for system administration tasks, managing database data, as well as writing GUI applications. One problem with the pearl language is its flexibility / complexity that makes it easier to write code that another author / coder has a hard time reading. (Rice's Theorem, 2008; Wikipedia, 2010)
**ASP.NET:** The Active Server Page (ASP.NET) was co-developed by Mark Anders, a manager on the IIS (Internet Information Server) team, and Scott Guthrie, who had joined Microsoft in 1997 after graduating from Duke University. The initial design was developed over the course of two months by Anders and Guthrie, and Guthrie coded the initial prototypes during the Christmas holidays in 1997. ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language such as VB.NET, C#, VC++.NET, etc. ASP.NET pages, known officially as "web forms", are the main building block for application development. Web forms are contained in files with an "aspx" extension; in programming jargon, these files typically contain static (X)HTML markup, as well as markup defining server-side Web Controls and User Controls where the developers place all the required static and dynamic content for the web page. Additionally, dynamic code which runs on the server can be embedded within webpages within a block <# --> dynamic code -- #> which is similar to other web development technologies such as PHP, JSP, etc. The biggest drawback of ASP is that it's a proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits it's availability to Win32 based servers. (http://www.asp.net/Retrieved, 2010)

**JavaScript:** JavaScript is an object-oriented scripting used to enable programmatic access to objects within both the client application and other applications. It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interfaces and dynamic websites. JavaScript is a dialect of the ECMA Script standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with. JavaScript was originally developed by Brendan Eich of Netscape under the name Mocha, which was later renamed to LiveScript, and finally to JavaScript. The change of name from LiveScript to JavaScript roughly coincided with Netscape adding support for Java technology in its Netscape Navigator web browser. JavaScript was first introduced and deployed in the Netscape browser version 2.0B3 in December 1995. The naming has caused confusion, giving the impression that the language is a spin-off of Java, and it has been characterized by many as a marketing ploy by Netscape to give JavaScript the cachet of what was then the hot new web-programming language. JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C, and JavaScript copies many Java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape bundling Sun's Java runtime with their then-dominant browser. The key design principles within JavaScript are inherited from the Smalltalk and Scheme programming languages.

**VBScript:** Visual Basic Scripting (VBScript) is an Active Scripting language, developed by Microsoft, which uses the Component Object Model to access elements of the environment within which it is running (e.g. FileSystemObject or FSO used to create, read, update and delete files). The language's syntax reflects its origins as a limited variation of Microsoft's Visual Basic programming language. VBScript has been installed by default in every desktop release of Microsoft Windows since Windows 98; as part of Windows Server since Windows NT 4.0 Option Pack; and optionally with Windows CE (depending on the device it is installed on). VBScript script must be executed within a host environment, of which there are several provided with Microsoft Windows, including: Windows Script Host (WSH), Internet (IE), Internet Information (IIS). Additionally, The VBScript hosting environment is embeddable in other programs, through technologies such as the Microsoft Script control. VBScript began as part of the Microsoft Windows Script Technologies, which were launched in 1996, initially targeted at web developers. During a period of just over two years, the VBScript and JScript languages advanced from version 1.0 to 2.0, and over that time it gained support from Windows system administrators seeing an automation tool more powerful than the batch language first developed in the late 1970s. In version 5.0, the functionality of VBScript was increased with new features such as: regular expressions; classes; the With statement; the Eval, Execute, and ExecuteGlobal functions to evaluate and execute script commands built during the execution of another script; a function-pointer system via GetRef, and Distributed COM (DCOM) support. In version 5.5, SubMatches were added to the regular expression class in VBScript, to finally allow VBScript script authors to capture the text within the expression's groups. That capability before was only possible through JScript. With the advent of the .NET framework, the scripting team took the decision to implement future support for VBScript within ASP.NET for web development, and therefore no new versions of the VBScript engine would be developed and it moved over to being supported by Microsoft's Sustaining Engineering Team, who are responsible for bug fixes and security enhancements. For Windows system administrators, Microsoft suggests that they migrate to Windows PowerShell. However the scripting engine will continue to be shipped with future releases of Microsoft Windows and IIS.

**PYTHON:** Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The programming language was conceived in the late 1980s by Guido Van Rossum at CWI in the Netherlands as a
successor to the ABC programming language (itself inspired by the SETL). Ever since, various versions of the language has emerged with new features ported with the various versions of the language. Python is a multi-paradigm programming language. This means that, rather than forcing programmers to adopt a particular style of programming, it permits several programming style: object oriented and structured programming are fully supported. The python language has a very good memory management capability. http://www.python.org/about/ (Retrieved, 2009)

**JSP**: Java Server Pages (JSP) is a Java technology designed by Sun Microsystems that allows software developers to create dynamically-generated web sites, with HTML, XML, or other document types, in response to a Web client request. The technology allows Java code and certain predefined actions to be embedded into static content. The JSP syntax adds additional XML-like tags, called JSP actions, to be used to invoke built-in functionality. Additionally, the technology allows for the creation of JSP tag libraries that act as extensions to the standard HTML or XML tags. Tag libraries provide a platform independent way of extending the capabilities of a Web server. The JSP engine/compiler (An engine that compiles codes written in the java server pages) is built around the Servlet Engine i.e. JSP is servlet made easy as most of the complicated task in servlet were made easy. JSPs are compiled into Java Servlets by a JSP compiler. Which may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet directly. JSPs can also be interpreted on-the-fly, reducing the time taken to reload changes. Even though JSP is platform

1. **PROGRAM LENGTH**

Program length is the same as the number of lines of codes present in the program which contains anything that contributes to the semantics of the program in each of the program files e.g., a statement, a declaration or at least a delimiter such as a closing brace or tags.

Length: manually counting the number of executable lines of codes for the implementation.

2. **Program Reliability**

It is concerned with how well programs behave. Web development tool reliability entails that the tool must consider unforeseen errors like syntax errors and other forms of language violations and responds appropriately by informing the user(s) of any violation instead of terminating executions inadvertently.

*Reliability:* measured by Web page output presentation in response of the Web browser / Web server.

3. **Development Cost**

The cost of developing a program in the target Web development tool is determined by the total man hours used in bringing out the final version of a Web application program, the cost in terms of the systems resources used and procurement of software and any other device or resource. *Development cost:* would involve a collection of all expenses incurred in getting each of the selected Web tools, the machine time utilized in terms of web browser

**II. MATERIALS AND METHODS**

Very often technology based decisions are made by technical personnel who base their decision on personal use, attendance at vendor sponsored workshops, reading about it in trade publications or having used other products from the same vendor. In this work, the empirical approach was used to evaluate the Web development tools under consideration, coding, compiling / interpreting and running programs in each Web tools.

1) **Ease Of Use**

Is the ease which the language is used in developing an application and the availability of structures that reduce programming complexity. For instance, some Web development tools with support for GUI are more users friendly and aid usability than Web tools without those features.

*Ease of use:* would be determined by taking note of how easier it is to write or design programs using any of the selected Web tools.

2) **Speed Of Execution**

The time it takes to compile and execute. The amount of time is measured by use of a stopwatch or by building in some program segment(s) to keep track of the execution and compilation time. *Speed of execution:* this is obtained by a program module that records start and stop time of execution.

3) **Platform**

This described the ability of software to run on a variety of different operating systems or the same operating systems. Different operating systems provide different platform challenges for web development tools. Issues with respect to 32 bit and 64 bit operating system are prominent. *Platform:* the ability of the program to run on the same or different operating systems and hardware.

4) **Functionality**

This described the ability of software to function properly or meet users’ needs in order to achieve their desired goals. *Functionality:* entails that the Web tools respond to users’ need in order to achieve their desire goals without any delay in delivery the Web content.

5) **Performance**

This described how will the Web development technologies can be used to achieve quickly and efficient delivery of applications. *Performance:* can be determined by considering how quickly and efficiently the selected tools are used to implement a Web site.
Functionality: entails that the Web tools respond to users’ need in order to achieve their desire goals without any delay in delivery the Web content.

5) Performance
This described how will the Web development technologies can be used to achieve quickly and efficient delivery of applications. 
Performance: can be determined by considering how quickly and efficiently the selected tools are used to implement a Web site.

6) Maintainability
This described how easy the Web development tools could adapt to changes when the need arises.
Maintainability: can be determining by considering how debugging is carried out when there is an error in the program and modifying to suit required upgrade.

7) Object Oriented Programming Design Facilities
This described the ability for the Web development tools to support OOP, which enhances reuse of object and quicker way to develop application.
Object-oriented programming facilities: determined by considering the various Web tools if they have the ability to used objects for programming reusability.

8) Database Supports
This described how the Web development technologies support a wide variety of back-end databases for effective records keeping.
Database supports: tests compatibility by linking the Web tools to a variety of database programs.

9) Portability
A term applied to software that is not dependent on the properties of a particular machine, and can therefore be used on any machine. Such software is also described as portable.
Portability: can be determined by implementing these tools on various computer machines in order to know whether it is machine dependent or independent

III. RESULTS AND DISCUSSION
Empirical Evaluation of Web development tools
Using the identified 12 criteria to evaluate the algorithm of each tool to ascertain their worthiness based on application developed with them. A simple shopping cart program was implemented with the case tools and the results obtained alongside the apparent conclusions are given as follows:

1) Program Length
The number of lines of code for each algorithm implemented for Web development tool is as shown below:

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTHON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable number of line of codes</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Average Program Length of the different Web development technologies
An examination of the data in Table 1 shows an increase in the source codes.

2) Program Reliability
Reliability of programs written in any programming languages/Web development technologies is never easy to determine as there are different parameters used by different software practitioners. In our own case, we considered the ease with which each of the Web development tools implemented their source code, their response when no input was fed in and the response of the interpreter to syntax errors. HTML and PHP were very reliable while the others were reliable.
3) Development Costs

The combined cost of acquiring an interpreter, setting it up, and that of the systems resources taken up by the Web page design in the target Web development technologies are summarized below in Table 3.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average cost of interpreter ($)</td>
<td>Free</td>
<td>Free</td>
<td>80</td>
<td>300</td>
<td>400</td>
<td>350</td>
<td>180</td>
<td>160</td>
<td>Free</td>
</tr>
<tr>
<td>Average set up costs ($)</td>
<td>Free</td>
<td>Free</td>
<td>2.40</td>
<td>4.50</td>
<td>6.80</td>
<td>3.20</td>
<td>2.70</td>
<td>2.64</td>
<td>5.40</td>
</tr>
<tr>
<td>Systems Requirements</td>
<td>1,064,356</td>
<td>1,024,543</td>
<td>1,004,564</td>
<td>1,324,097</td>
<td>1,423,206</td>
<td>1,300,340</td>
<td>1,320,543</td>
<td>1,375,300</td>
<td>1,400,200</td>
</tr>
<tr>
<td>Costs of writing Programs/computer time</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
</tr>
</tbody>
</table>

Table 3: Development Costs of the different Web development technologies

4) Ease Of Use

Rating the easy to use of the Web development technology with a scale of 1 to 2 (where 1 denotes easiest to learn and 2 denote easy). Table 4 shows that HTML, PHP, CF, PERL, and JSP are the easiest to learn, while ASP, JS, VBS and PYTON are easy to use.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use (When scaled)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: Ease of Use of the selected Web development technologies

5) Speed Of Execution

The speed of execution (measured in seconds) for the simple Web page created is given below in tables.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpreting/speed of execution (in seconds)</td>
<td>0.56</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.58</td>
<td>0.58</td>
<td>0.58</td>
<td>0.58</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 5: Average Speed of Execution of the different Web development technologies
6) **Platform**
Testing each tool on two major operating systems; Windows Operating System 98 and Windows XP Operating system determined the platform supports by the Web development tools examined. From our observation, it was apparent to us, to scale the tools into the platform they support: 1 for dependent; 2 for independent. Table 6 shows that all the programming languages are independent.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform (When scaled)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 6: Platform Support of the different Web development technologies

7) **Functionality**
Functionality of the Web development tools can measure by the way the tools meets users need based on the fact that they can easily be used to designed sophisticated Web page or Web application. Thus, the functionality can also be rated as 1 for excellent, 2 for very good, 3 for good, and 4 for poor.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality (When scaled)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 7: Functionality of the different Web development technologies

8) **Maintainability**
Maintainability had to be measured by use of established methods for determining the effect of modifying or debugging of errors from the coded Web development tools. It was observed that HTML, PHP, CF, and PERL were very easy to maintain while ASP, JS, VBS, PYTON, and JSP were easy to maintain. From the general point of view, we can conclude that the maintainability of Web tools is okay.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintainability (When scaled)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 8: Maintainability of the different Web development technologies

9) **Object Oriented Programming Facilities**
This can be determined by the usage of its features in the various Web development technologies. From our observation, we can rate the various tools into the classes they belong. That is, 1 for excellent, 2 for very good, 3 for good, and 4 for poor.

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOP (When scaled)</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 9: OOP Facilities of the different Web development technologies

Determining the development costs for the various language implementations was a bit difficult and we had to arrive at a compromise to base the actual costs of the interpreters in US Dollars since that is the most recognized currency used in international and on-line business transactions. The costs of the interpreters were found to vary from one marketer to the other and after comparing prices for 8 (eight) different retailers, we took the average price which was in the range of prices offered and which included the prices for shipping. The set up costs indicated in the offer prices were used while for the cost of writing, coding and running each algorithm in the selected languages, a flat rate of $1.50 per hour of computer time was assumed if one were to carry out the programming task in a commercial center. The assumed price is closest to the N250 charged per hour of computer...
time in business centers. An examination of table 9 shows that the development cost of the HTML program is the cheapest while the development costs of the Macromedia ColdFusion program is the most expensive.

10) Database Supports

The database supports is determined by the linkage of the Web page to some of the various database software. Form our observation, the following rating was used to determined the levy of support to the database, that is, (1 for excellent, 2 for very good, 3 for good, and 4 for poor

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database supports (When scaled)</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 10: Database supports of the different Web development technologies

11) Portability

The portability of the Web development tools based on the simple Web page design which then was measured based on the execution of the program in different computer machines. From our observation, we can rate the portability into scale as (1 for highly portable, 2 for portable, 3 for fairly portable and 4 for not portable).

<table>
<thead>
<tr>
<th>Technology</th>
<th>HTML</th>
<th>PHP</th>
<th>CF</th>
<th>PERL</th>
<th>ASP</th>
<th>JS</th>
<th>VBS</th>
<th>PYTON</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portability (When scaled)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 11: Portability of the different Web development technologies

IV. CONCLUSION

From our findings in this research, we conclude that the choice of users to a particular web tool is based on the task at hand which may be designing a simple webpage:— Use HTML, FrontPage editor or PHP; designing for efficient database support:— PHP, CF, ASP, JS, VBS, and PYTON; designing for Object oriented task:— Use PHP, Java Script; designing for Low cost:— PHP, HTML, ASP, VBS, JS; designing for small program length:— JSP since it requires less coding; designing for portability:— PHP, HTML, PYTON, and JS; designing for functionality:— PHP, JSP and XML and designing for Speed of Execution:— PHP. This shows that the use of reliable approaches or software development processes, copious analyst, good designers, and painstaking implementation techniques are a prerequisite. Thus, very rich design and coded Web site for transactions in the Internet produces attractiveness, and user-friendly Web pages.

V. RECOMMENDATION

Web development technologies have become the core of most organization, Corporation, Research Institute, Government presence on the Internet, it is imperative that the use of these tools be accorded the attention it desires. One way of according this attention is for organization, corporate companies, research institutions and Government to acquire these sophisticated tools and trained their IT staff on how to use the tools to develop dynamic and interactive Web sites or custom applications that will ease this function. We strongly recommend that individual or organizations should select their Web development tools based on the level of usability, which strongly encompasses other factors. Universities do not just produce graduates strictly for academic research but also for industrial purposes, therefore, there is the need for Lecturers to be well acquainted with these tools in order to empower their students for job opportunities after graduation

Who will benefit from this report?

IT managers and strategists who need to assess whether they should develop business applications for the Web, and if so, when and how to do it

Software developers who need to identify the right tool for their projects

Business managers who need to understand how the Web can make their business processes simpler and more efficient

Application development tool vendors who need to identify opportunities for partnership or assess competing products

Consultants and systems integrators who need to advise their clients on the variety of different approaches to intranet and WWW development

VI. REFERENCES


10) Getting started with Perl and CGI available at http://www.perl.com/cgi


12) Getting started with JavaScript available at http://www.javaScript.com


16) Introduction to XHTML available at http://www.w3schools.com/xhtml/default.asp


21) Macromedia Flash MX repository available at http://www.macromedia.com/software/flash


34) Web building tutorials repository available at http://www.w3schools.com


36) XML repository available at http://www.xml.com