Establishing and Applying Criteria for Evaluating the Ease of Use of Dynamic Platforms for Teaching Web Application Development

Johnson Dehinbo
Dehinbooj@tut.ac.za
Department of Web & Multimedia Computing,
Tshwane University of Technology, Pretoria, South Africa

Abstract

The widespread use of the Internet and the World Wide Web led to the availability of many platforms for developing dynamic Web application and the problem of choosing the most appropriate platform that will be easy to use for undergraduate students of web applications development in tertiary institutions. Students beginning to learn web application development would not perform at their best capacity level if the platform chosen by their institution is difficult to use. This is important given the recurrent write-compile-test-recompile efforts that take place in a timed students' practice session. As part of the framework for determining the most suitable platform(s) for teaching dynamic web applications development in tertiary institutions, this study adopted an interpretive research approach to establish a set of criteria from theory and practice for evaluating the ease of use of the platforms. These criteria were tested by evaluating four platforms namely Java Servlets, Java Server Pages (JSP), Active Server Pages (ASP) and PHP using various research methods including descriptive inquiry, document analysis and observations. While Java Servlet was found to be most suitable on applying the criteria, the significance of the study lies in the establishment of a comprehensive but specific set of criteria that can be used as a scientific basis for selection.

Keywords: Web application development, programming, languages, platforms, usability, ease of use

1. INTRODUCTION

This study established part of a framework containing various criteria that can be used to evaluate dynamic Web application development platforms in terms of their suitability for teaching Web applications development in tertiary institutions. This part of the framework determines how to choose the platform that would be easiest to use by undergraduate students. The aim of the study was not to persuade readers that one platform is better than another, but to help readers make an informed decision on the ease use of the platforms.

There are currently many platforms for developing dynamic application (that are interactive, back-end database enabled, session-enabled, and with processing capabilities) on the World Wide Web. According to Lim (2002) information systems/ computer science departments need to reexamine their curricula in order to prepare students to face the challenge of being productive in a computing world swamped with web technologies. We believe that the productivity of the students would be enhanced by the choice of a platform that is easy to use in their web application development efforts. The choice made may affect the speed and efficiency with which students can develop the web applications. Web application development students would not perform at their best capacity with a platform that is difficult to utilize.

A good way of teaching Web application development would be to teach using a platform

that is suitable for the students. The problem however is how best to make a reasonable choice of platform from the various platforms. It is assumed that this would involve analysis and evaluation in terms of various factors in line with the idea put forward by Ashenfelter in the statement below:

> Web development tools need to be analyzed in terms of its purpose (what it is designed to do), technology (ease of use, robustness, scalability, security, performance, etc.), support (portability, cost, ISP support), and how well it works in the real world. (Ashenfelter, 1999:109)

Thus, by implication, ease of use is one of the critical factors to be taken into consideration in choosing the platform to be used. The research question then is: How could one scientifically determine the dynamic web application development platform that will be easy to use for undergraduate students in tertiary institutions? This is in line with Ashenfelter's (1999:105) assertion that before analyzing tools, it is worth discussing how to evaluate them.

The objectives of the study are therefore:
1. To establish criteria to investigate the ease of use of dynamic web application development platforms;
2. To apply the established criteria to evaluate specific web application development platforms.

The main benefit of the study will be the educative analysis and evaluation in choosing suitable dynamic web application development platform to enhance the performance of learners, thereby increasing their potentials in such a way that would lead to higher productivity for the students.

The rest of the paper is structured as follows. Section 2 discusses the literature review highlighting the uniqueness of the study. Section 3 presents the research approach and methodology. The criteria were established in section 4 and applied in section 5. Section 6 concludes the study.

2. LITERATURE REVIEW

Analyzing, evaluating and comparing programming languages, development platforms

and tools are important but also difficult as illustrated in the following statement:

> Comparisons across programming styles, or paradigms, are difficult to carry out, but are nevertheless important for understanding how different styles of programming affect the learning of novice programmers. (Wiedenbeck et al., 1999:105)

Apte *et al.* (2003) note that a study of existing literature showed varying conclusions about the superiority of one dynamic Web platform over another. Prechelt (2000) indicates that when it comes to the advantages and disadvantages of various programming languages and development platforms, programmers and computer scientists alike usually hold strong and different opinions. This is evident in the different approaches to the comparisons and selections of programming languages and platforms in the literature ranging from those based simply on intuition to those based on comprehensive criteria.

However, most of these studies did not use any explicit criterion as a basis for their comparisons, but are based on intuition rather than scientific facts. This study however believes that scientific evidence is required to support the various assertions. Comparison should be based on a variety of factors supported by scientific facts relevant literatures, experience and empirical observation results. This would involve using specific factors or criteria in the comparisons as in the next section.

Towards incorporating various criteria in their comparisons

Vinoski (2003) realizes that various comparisons of programming languages and platforms concentrate on performance comparisons. These include Renaud *et al* (2003), Cooper (2001) as well as Marshak and Levy (2003). A suitable framework for comparison should involve other relevant factors. Vinoski (2003) therefore shows that, depending on the application, other qualities such as scalability, flexibility and adaptability, ease of use, tool support, and standards conformance could very well take precedence over performance.

This study therefore proposes that the advantages and strengths of each platform should be examined in the light of certain desired qualities relevant to the criteria of interest to a specific use. For example, the ease

of use of a platform would increase productivity in institutions training middle-level workers than the flexibility of the platform. It is on this note that it is deemed necessary to focus on criteria of interest towards enhancing the ease of use of the platforms by undergraduates. This leads to the review various studies on usability as presented next.

Towards a framework for usability estimation of web platforms
The ease of use of platforms for teaching web applications development is closely related to the concept of usability, except that here the focus is on knowing how usable the tools used to design the software products are, rather than the usability of a designed application software product. According to Preece *et al.* (2002) cited in Singh and Kotze (2002), usability is generally regarded as ensuring that interactive applications are easy to learn, effective to use, enjoyable, and involve the optimization of user interaction with these products.

Ashenfelter (1999:111) notes that Web databases require a knowledge that spans the worlds of Webmaster, database administrator, and programmer. Therefore, learning to use dynamic Web platforms is relatively difficult compared to other programming languages. However, the ease with which new tools can be learned has an impact on how much they will be used. Ashenfelter (1999:112) identifies a number of factors that can affect how easily a Web platform can be learned. Firstly, unnecessarily complex platforms are the biggest obstacle to ease of learning and thereby affecting ease of use. Secondly, familiarity of the related components of the dynamic Web platform will enhance learning. Thirdly, availability of useful documentation will be of great assistance in learning and using the platform.

Singh *et al.* (2001) states that several researchers have over the years produced sets of principles or guidelines aimed at improving the usability of interactive systems. These as listed in Singh and Kotze (2002) include aspects such as effectiveness, efficiency, safety, utility, "*learnability*", flexibility, robustness, "*memorability*", *etc*.

Adeyemo (2000) states that on the use of applications, it really looks like "Usability or else!" for the business. Although Adeyemo focused on ease of navigation of websites, some possible usability measurement criteria adapted from Keller (1990:287) were listed, which include percentage of task completed per unit time (speed metric); ratio of successes to failures; time spent in errors; percentage of competitors that do this better than current product; number of commands/statements used; frequency of help or documentation use; time spent using help or documentation; percentage of favorable or unfavorable comments; number of good and bad features recalled by user; number of available commands not invoked; number of times the user need to work around a problem; number of times the user expresses frustration or satisfaction etc.

Wesson and Van Greunen (2002) summarized usability data as comprising the three aspects of the definition of usability: efficiency, effectiveness and user satisfaction: Effectiveness relates the goals of using the product to the accuracy and completeness with which these goals can be achieved. Common measures of effectiveness include the percentage task completion, frequency of errors, frequency of assists and frequency of access to help or documentation (Wesson & Van Greunen, 2002); Efficiency relates to the level of effectiveness achieved to the quantity of resources expended. Efficiency or performance is generally assessed by the mean time taken to achieve the task; Satisfaction or acceptance describes a user's subjective level of satisfaction when using the product (Wesson & Van Greunen, 2002).

Dix *et al*. (1998) put forward principles to support usability in three categories: "Learnability", flexibility and robustness. "Learnability" refers to the ease with which new users can begin effective interaction and then attain a maximal level of performance. Usability principles related to learnability include predictability, synthesizability, familiarity, generalizability and consistency. Flexibility refers to the multiplicity of ways the user and the system exchange information. Usability principles related to flexibility include dialogue initiative, multi-threading, task "migratability", "substitutivity", and "customizability". Robustness refers to the level of support provided to the user in determining successful achievement and assessment of goals. Usability principles related to robustness include 'observability', recoverability, responsiveness and task conformance.

However, for dynamic web application systems, the focus of usability is on the features of the programming language or web platform being used in the development of the system. Thus, in summary, we could think of evaluating usability as necessary for the following reasons adapted from (Adeyemo,2000): educating other computer professionals; increasing satisfaction; decreasing anxiety (which server should I use, which database? etc); increasing trust of technology; increasing consistency across products; decreasing learning time- how long does it take for typical designers to learn to use the platform; increasing performance - how long does it take to perform tasks and how many and what kinds of errors are commonly made during typical use; increasing retention over time; increasing productivity- how many source line of codes (SLC) can be written per man-month and how easy to understand error messages and how long does it take to work around the error; establishing scalability - adaptability to growth in size of a system e.g. increased users or large databases; good development environment and validation tools; availability of add-ons e.g. email; support for software reuse and maintenance; security of information *etc*.

These metrics provide a measure of usability against which the platforms can be tested. However, since no single platform will likely positively satisfy all the criteria, there is the need for a systematic way to determine which platform will satisfy a greater number of the vital criteria. This leads us to the research design and methodology presented in the next section.

3. RESEARCH DESIGN AND METHODOLOGY

An interpretive research approach is used involving elements of descriptive, analytical and comparative studies. McMillan and Schumacher (2001: 33) state that while a descriptive study describes a system with the aim of characterizing it as it is, by using numbers, comparative study investigates the differences, thereby taking descriptive study a step further. A descriptive approach is used to characterize desirable features of the platforms towards establishing a set of criteria. In applying the established criteria, an analytical approach is used to analyze and evaluate platforms to determine the level of satisfaction of the criteria in order to compare the platforms with the goal of determining their ease of use towards choosing the most suitable platform.

Research methods adopted to obtain the results

The descriptive method used involves document review and study of various manuals and textbooks for different platforms from various sources and established body of knowledge to identify features that could enhance the ease of use of platforms for web application development. These features were characterized to establish criteria for evaluating the ease of use of platforms by undergraduate students. The established criteria serve as a model. Bowling (2002:141) describes models as abstract representations of the essential characteristics of phenomena of interest that make explicit the relationships and comparison between the characteristics. With the model in place, the stage is set for application of the criteria by analyzing, evaluating and comparing specific platforms.

The analytical method is aimed at evaluating specific platforms to ascertain their level of satisfaction of the established criteria. Documents were reviewed for the specific platforms to be evaluated and compared using various sources and established bodies of knowledge. Moreover, there is the need to physically examine the ease of use of the dynamic web application development platforms. Bowling (2002:358) describes observation as a research method in which the investigator systematically watches, listens to and records the phenomenon of interest. In this study, the use of observation was limited to the inquiry into the features and facilities available on the installed platforms and their corresponding Integrated Development Environments (IDEs) as well as on observations from experience on the use of features of the platforms for web application development.

Answers were sought to the questions and the availability of features that serve as the criteria from established sources which include those written by the designers and originators of the platforms. These were augmented by authoritative websites including those of Sun Microsystems and Microsoft as well as PHP websites and various websites for the applicable web servers such as Internet Information Service (IIS) and tomcat. Scores were assigned to each platform based on the availability of desired features, the level of support enabled on specific tasks or the inherent characteristic of the platform such as readability of program codes.

Measuring scale used
Using close-ended "Yes/No" questions, the measuring tool has values on a scale of 1 to 3, where:

> **3 = "Yes",**
> **2 = "Not quite or with some workaround", and**
> **1= "No".**

A scale of 1 to 3 is used to avoid subjective situations where it will be difficult to distinguish between, for example, a score of 3 or 4 in a scale of 1 to 5. The use of the 1 to 3 scale therefore reduces the situation as to whether or not the facility is available, or in between.

Establishing reliability and validity
A measuring scale or instrument should be consistent and reliable. It should produce more or less the same accurate results every time it is applied, even when applied by different persons (Coertze & Heath, 1997: 78). Coertze and Heath (1997: 79) indicate that validity is concerned with soundness or the effectiveness of the measuring instrument. As a way of increasing validity, answers to the criteria questions were sought from recognized textbooks, authoritative websites, and journal articles. We provided accompanying references so that interested readers can verify or seek more information.

Also, to increase reliability, the quantitative characterization and evaluation using numbers would enhance clarity in the choice of platform with the highest score. This is unlike just using qualitative sentences to evaluate the platforms, at the end of which it would be difficult to say which platform is really more suitable. Furthermore, the avoidance of subjectivity by the range of values is aimed at enhancing reliability.

Data analysis
Qualitative analysis was used in the establishment of the criteria as well as in the characterization involved in the application of the criteria. Simple parametric analysis involving means was later used in selecting the most suitable platform. The scores for all the criteria were summed up for each web-based dynamic platform to obtain a total score from which the platform with the highest overall score can be identified.

Limitations
A particular limitation of interpretive studies is that it gives a subjective view of the researcher.

The criteria established are based on sources of knowledge currently available to the researcher and the interpretation thereof which is also based on the researcher's experience. Therefore, the study viewed as a guide. While users can add or remove certain criteria, the next section presents the contribution to the body of knowledge in form of the criteria to evaluate the ease of use of dynamic Web application development platforms.

4. THE CRITERIA DEVELOPMENT

The establishment of the criteria to evaluate the ease of use of the platforms involves identifying qualities that would ensure that the platform will be easy to use for students studying Web application development. Therefore, the criteria should determine a platform with a user-friendly IDE. In addition, the platform should be less resource-intensive and also be portable. The desired qualities that will serve as criteria to be used to evaluate the ease of the Web application development platforms is now presented.

Availability of smart Integrated Development Environment that could enhance ease of use
This involves the availability of interpreters, compilers, as well as the availability of smart integrated editing system. According to Sebesta (1996:18), the effort of training programmers and developing applications can be significantly reduced with a good programming environment. A programming environment is the collection of tools used in the development. It may consist of only a file system, a text editor, a linker and a compiler or it may include a large collection of integrated tools such as a debugger, which when encounter error, stops and switches to the editor, leaving the cursor at the point in the program where the error was detected (Sebesta, 1996:31).

It is also important whether the interface to the tool is graphical, window-based and similar to common interfaces. Bergin (1996) states the need for a platform independent graphics model and a platform independent window-button-mouse based GUI.

The following criteria are considered necessary to ensure that our desired platform has a user-friendly and smart IDE:

> ☑ Availability of interpreters rather than compilers; for undergraduate students,

interpreters will be good for quickly executing what they have done so far, and for seeing the effects of minor changes to the programs.

☑ Availability of an IDE that is both DOS and Windows-based; this way, the system can easily be used as preferred.

☑ Availability of smart integrated editing systems that:

➢ Enhance maintenance by encouraging good readability of the codes, by way of automatically formatting and indenting codes.

➢ Are equipped with tracing and debugging aids to aid students in debugging their codes.

➢ Offer other useful and smart integrated editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time.

➢ Are equipped with context-sensitive "help options" to assist students who are 'stuck' in certain activities during their Web applications development.

➢ Have the ability to switch case sensitivity on and off so that it can be turned on when there is the need for many related identifiers, such that their minor differences can be represented by alternating the case of the letters of the identifiers.

5. SUMMARY OF THE RESULTS OF APPLYING THE CRITERIA

We applied the criteria to evaluate the ease of use of four platforms namely Java Servlets, JavaServer Pages (JSP), Active Server Pages (ASP) and Personal Home Page (PHP). The results are summarized below.

Availability of features of smart Integrated Development Environment
Table 1 and table 2 reflect the scoring for the dynamic Web platforms based on the criteria on the availability of features of smart IDEs. A summary of these results based on a review of various literatures, detailed observation and testing of various IDEs and from programming experience follows.

A compiler wouldn't present a result if the program is incomplete on contains an error. The use of an interpreter rather than a compiler could allow students to instantly see results of programs up to where error is encountered or up to what has been programmed so far, irrespective of whether the program is complete or not. Apart from Java Servlet in which one needs to compile the program, other platforms use interpreters such that the incremental building of the programs becomes easier for students. In the case of JSP, however, the interpretation involves internal compilation.

Also, for prompt testing and incremental building of the programs, it is better that the platform allows execution from both the DOS prompt and the browser. The Java Servlet allows execution from the DOS prompt through the use of the "Servlet runner" program (Sun Microsystems, 1997) and also from the browser. JSP and ASP are only executable via the browser. The PHP engine can be invoked as either a CGI-callable interpreter from supported UNIX platforms and Windows NT, or as an in-process Apache module that can be viewed from all HTML-capable browsers (Ross *et al.*, 2000). PHP can be used for server-side scripting via browser as well as command-line scripting via DOS prompt and Unix shell (Lerdorf & Tatroe, 2002:1).

It is important to know in which directory to put the scripts, HTML pages and graphics. The Java platforms' JCreator IDE automatically creates directories to store the project for the Java-based platforms (JCreator, 2008). The same goes for Microsoft Visual Interdev IDE for ASP (MSDN, 2008) and PhpEd for PHP (Yank, 2008). In addition, a look at all the IDEs shows that they allow menu-driven addition of Web items such as HTML pages, scripts and style sheets, which are put in the appropriate sub-directories.

Apart from creating the directory, the IDE should also show the directory tree for the project to enable the students to browse and modify files easily. This feature is available in JCreator for the Java based platforms (JCreator, 2008), and in Microsoft Visual Interdev IDE for ASP (MSDN, 2008), as well as in PhpEd for PHP (Yank, 2008).

It is important that the IDE enhances maintenance by encouraging good readability and writability by way of automatically formatting and indenting codes. JCreator as editor for Java Servlet and JSP programming automatically format and indent codes (JCreator, 2008). Similarly, Microsoft Visual InterDev, usable as editor for ASP codes, has formatting and indenting capabilities (MSDN, 2008). PHP's

PhpED editor also automatically formats and indents the codes (Yank, 2008).
Since parentheses and brackets are always in pairs, it is useful if the IDE has hotkey matching of braces, parentheses and angle brackets. Practical testing shows that the various IDEs such as JBuilder and JCreator for the Java-based platforms, as well as the Visual InterDev for ASP have this facility. PhpEd for PHP also has this facility (Yank, 2008).

Since there is usually a special format for the HTML files, scripts, *etc,* it is preferable if the IDE has customizable code formatting templates such that students can just add contents to the templates. This feature is available in JCreator (JCreator, 2008) and in Microsoft Visual Interdev IDE for ASP (MSDN, 2008) as well as in PhpEd for PHP (Yank, 2008).

An IDE that is equipped with tracing and debugging aids would assist students in debugging their codes. Examination of the various IDEs shows that they all have tracing and debugging features.

It is important that the IDE should offer other useful and smart editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time. Examination of the various IDEs shows that none have this feature.

Given that colors enhance visualization, it is of some importance that the IDE provides syntax highlighting with configurable colors. Practical testing of the various platforms' IDEs shows that they all have this feature.

Another desirable feature is the use of functions and variable name auto-completion which would eliminate the use of wrong names and also speed up the coding process. The IDEs for the Java-based platforms have this feature. Unlike the IDE for ASP, PhpEd for PHP also has this feature (Yank, 2008).

Since the IDE knows the functions applicable in a platform, it will be useful if it has pop-up parameter references for recognized functions. The IDEs for the Java-based platforms, as well as Microsoft Visual InterDev for ASP, have this feature. PhpEd for PHP has it as well (Yank, 2008).

In accordance with the move towards GUI-based systems, it could be better if the IDE has an HTML toolbox for adding HTML components as well as "What-You-See-Is-What-You-Get" (WYSIWIG) editing. Unlike the IDEs for the other platforms evaluated, only the Microsoft Visual InterDev for ASP features this (MSDN, 2008).

Given that most Web applications involve the use of database, teaching could become easier if the IDE enhances database integration by having database tools such as those for viewing tables, adding connection objects *etc*. Detailed examination of the IDEs shows that this feature is missing in the IDEs for the Java-based platforms, but available in Visual InterDev for ASP (MSDN, 2008), as well as in EasyPHP IDE for PHP. Fuecks (2008) confirms this by warning former VB users not to forget that there is ADOdb which is an ADO equivalent for PHP.

Furthermore, being equipped with context sensitive helps could enable such IDEs to assist students who are "stuck" in certain programming activities. Examination of the various IDEs shows that those for the Java-based platforms have such features, unlike the IDEs for the other platforms evaluated.

Detailed information could be easily accessed if the IDE allows searching documentation and Application Programmers' Interface (API) from the help menu. All the IDEs for the various platforms have such features (JBuilder, 2008; MSDN, 2008; Yank, 2008).

In long statements, it is useful if the IDE shows split views of the long scripts instead of requiring one to scroll and lose sight of beginning portions. It is observed that this feature was missing in the IDEs for the Java based platforms and ASP. PhpEd for PHP does, however, have this feature (Yank, 2008).

While coding and testing developed systems, one often needs to see different views such as the class view, program view, debugging view, and the output view. It is observed that all the IDEs for the various platforms have this feature.

For those functions obtainable via the menu, it would be additionally beneficial if the IDE has configurable keyboard shortcuts. All the IDEs for the various platforms have this feature.

Even after the script is syntactically correct, students often have difficulty in constructing the applicable Uniform Resource Locator (URL) to view the results. A useful feature is IDE's

automatic construction of the URL for the output at the click of a button. Unlike the IDE for the Java-based platforms, observation shows that this feature is available in the IDEs for ASP and PHP.

Although it was previously established that it would be useful for novices if the platform is case-insensitive, it would be even better if the IDE has the ability to switch case-sensitivity on and off as necessary. Observation shows that the IDEs for the Java-based platforms have such feature (JCreator, 2008), unlike the IDEs for the other platforms under study.

If a platform is equipped with an independent GUI development tool, a situation where one has to learn and use another language for the front-end design can be avoided. The Java-based platforms have the "Swing" class for the front-end GUI design, unlike the IDEs for the other platforms that use HTML for the front-end GUI design.

Also useful is a *class* wizard which assists in the creation of classes. JBuilder has such facility (JBuilder, 2008) but the other platforms do not.

It is easier to be able to start and stop the server from within the IDE. Only PhpEd for PHP has such facility.

Last but not the least, it would be valuable if an IDE supports an unlimited undo and redo feature so that desired actions can be repeated and unwanted changes can be easily discarded. All the IDEs for the platforms under consideration have 'undo' and 'redo' buttons (JBuilder, 2008; MSDN, 2008; Yank, 2008).

Summary

The application of the established criteria reveals that Java Servlet has the highest score followed by JSP and then ASP and PHP. Although ASP has simpler architecture with good IDE, its drawbacks stem from being largely Microsoft-based, costly and less able to interoperate with other platforms. It should however be noted that the scores are subjectively based on the researcher's knowledge and experience as well as the current design of the platforms. They are thus subject to change with time. Therefore, the emphasis of this study is on the establishment of the criteria that enables the selection among the platforms. Users can thus adapt these criteria to their own taste, associate weights to each criterion and can also test or apply the criteria on other platforms.

6. CONCLUSION

It is important to ensure that the various tasks in web application development are done with adequate ease and in reasonable time. These ideas form the basis of this study to evaluate the ease of use of dynamic web application development platforms. By studying the features of many platforms, features that are desirable for enhancing the ease of use of Web application development platforms are identified. These enabled the establishment of criteria to determine the ease of use of the platforms. The significance of the study lies in the establishment of a comprehensive but specific set of criteria that can be used as a scientific basis for selection. Evaluation of platforms with this framework will contribute towards the determination of suitable platforms for teaching web application development in tertiary institutions.

7. REFERENCES

Adeyemo, A.M. (2000) "Towards a framework for the evaluation of Web sites", *M.Com. dissertation*, University of Pretoria.

Apte, V., Hansen, T. and Reeser, P. (2003) "Performance comparison of dynamic Web platforms", *Computer Communications*. Vol 26, No.8, pp888–898.

Ashenfelter, J.P. (1999) *Choosing a Database for Your Web Site.* John Wiley & Sons, New York.

Bergin, J. (1996) "Object Technology in the Classroom-Java as a Better C++", *ACM SIGPLAN Curricular Patterns,* Vol 1, No.2, pp21-27.

Bowling, A. (2002) *Research methods in health.* 2nd edition, Open University Press, Buckingham.

Cecchet, E., Chanda, A., Elnikety, S., Marguerite, J. and Zwanenpoel, W. (2003). "Performance Comparison of Middleware Architectures for Generating Dynamic Web Content", *Lecture Notes in Computer Science: Middleware,* Vol 2672, No.2003, pp242-261.

Coertze, D. and Heath, R. (1997) *Research Methodology for Technikon Students: A*

practical Approach. Technikon Natal publishing, South Africa.

Cooper, R. (2001) "Software for managing Web sites" In: *Proceedings of the South African Institute of Computer Scientists and Information Technologists (SAICSIT) Annual conference*. September 2001. Pretoria, South Africa: SAICSIT.

Devdaily. (2009). *Servlet Runner 101: Testing your Java Servlets with the servletrunner* [online], http://www.devdaily.com/java/edu/pj/pj010025/index.shtml

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998) *Human-Computer Interaction*, Prentice Hall, Hertfordshire.

Fuecks, H. *Is PHP the natural upgrade path for ASP 3.0/VB 6?* [online], http://www.sitepoint.com/blog-post-view.php?id=175686

Hartman, H. (2001). Tools for dynamic Web sites: ASP vs PHP vs ASP.NET, *Seybold Report Analysing Publishing Technologies*, Vol 15339211, No.12, pp1-10.

JBuilder. (2008) "*JBuilder 9 Personal Tutorial*", [online], http://www.cs.dal/~arc/teaching/CS2110/TutorialS/introToJBuilderTutorial.htm

Jcreator. (2008) "*Using JCreator: IDE tutorial*", [online],http://www.cs.usm.maine.edu/~welty/cos160JCreatorOnlineTutorial/

Lerdorf, R. and Tatroe, K. (2002) *Programming PHP,* O'Reilly & Associates Inc, Sebastopol, Calif.

Lim, B.L. (2002) "Teaching web development technologies: Past, present, and (near) future", *Journal of Information Systems Education*. Vol 13, No.2, pp117-123.

Marshak, M. and Levy, H. (2003) "Evaluating Web user perceived latency using server side measurements", *Computer Communications, Vol* 26, No.8, pp872-887.

McMillan, J.H. and Schumacher, S. (2001) *Research in education*. 5th edition, Addison Wesley Longman, New York.

MSDN. (2008) "Introducing Visual InterDev", *Microsoft Visual Studio Help file*.

Prechelt, L. (2000) "An empirical comparison of seven programming languages", *Computer*. Vol 33, No.10, pp23-29.

Ross, D., Zymaris, A. and Con, D. (2000) "DB Forms, PHP, MYSQL, and PHPLIB: A reusable database framework for writing Web apps", *Dr. Dobb's Journal: Software tools for the professional programmer*. Vol 25, No.8, pp11-16.

Sebesta, R.W. (1996) *Concepts of Programming Languages,* Addison-Wesley, Reading, Mass.

Singh, S., Erwin, G. and Kotze, P. (2001) "Electronic Business Accepted Practices (e-BAP): Standardised HCI for e-commerce in South Africa", In: *Proceedings of the SAICSIT Annual Postgraduate research symposium*. September 2001. Pretoria, South Africa.

Singh, S. and Kotze, P. (2002) "Towards a framework for e-commerce usability", In: *Proceedings of the annual SAICSIT conference*. September 2002. Port Elizabeth, South Africa.

Sun Microsystems. (1997) "JDBC Guide: Getting Started", [online], http://www.pearson.com.mx/component/component_java_21_days/jdk/1.2/doc/html/guide/jdbc/getstart/introTOC.doc.html

Wiedenbeck, S., Ramalingam, V., Sarasamma, S. and Corritore, C.L. (1999) "A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*", Vol 11, No.3, pp252-282.

Yank, K. (2008) "Review – PHPEd 3.2 (NuSphere)", [online]. http://www.sitepoint.com/print/review-phped-3-2-nusphere

8. APPENDIX

Table 1. Scoring for the platforms based on the criteria on the availability of features of smart IDEs.

	Criteria questions	Servlet	JSP	ASP	PHP
1	Does the platform use interpreters rather than compilers?	1	2	3	3
2	Is the IDE both DOS and Window based?	3	1	1	3
3	Does the IDE automatically create directories to store the project?	3	3	3	1
4	Does the IDE show the directory tree for the project?	3	3	3	3
5	Does the IDE allow the menu-driven addition of Web items such as HTML pages, scripts, style sheets, etc.?	3	3	3	3
6	Does the IDE enhance maintenance by encouraging good readability and writability by way of automatically formatting and indenting codes?	3	3	2	3
7	Does the IDE have hotkey matching of braces, parenthesis, and angle brackets?	3	3	1	2
8	Does the IDE have customizable code formatting templates?	3	3	3	3
9	Is the IDE equipped with tracing and debugging aids to assist students in debugging their codes?	3	3	3	3
10	Does the IDE offer other useful and smart editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time?	1	1	1	1
11	Does the IDE give syntax highlighting with configurable colors?	3	3	3	3
12	Does the IDE have the ability to switch case sensitivity on and off as necessary?	3	3	1	1
13	Does the IDE have function and variable name auto-completion?	3	3	1	2

Scale: 3 = "Yes", 2 = "Not quite or with some workaround", and   1= "No".

Table 2. Scoring for the platforms based on the criteria on the availability of features of smart IDEs continued.

	Criteria questions	Servlet	JSP	ASP	PHP
14	Does the IDE have pop-up parameter references for recognized functions?	3	3	3	3
15	Does the IDE have an HTML toolbox for adding HTML components as well as WYSIWIG editing?	1	1	3	1
16	Does the IDE enhance database integration by having database tools such as viewing tables, adding connection objects?	1	1	2	2
17	Is the IDE equipped with context sensitive helps to assist students who are "stuck" in certain programming activities?	3	3	2	1
18	Does the IDE allow searching documentation and API from the help menu?	3	3	3	1
19	Does the IDE show split views of long scripts?	1	1	1	3
20	Does the IDE show different views such as the explorer class view, program view, debugging view, output view?	3	3	3	2
21	Does the IDE have configurable keyboard shortcuts?	3	3	3	3
22	Does the IDE automatically construct the URL for the output?	1	1	3	3
23	Is the platform equipped with an independent GUI development tool?	3	3	2	1
24	Is there a class wizard to assist in the creating of classes?	3	3	1	1
25	Is it possible to start and stop the server from within the IDE?	1	1	1	3
26	Does the IDE support unlimited undo and redo?	3	3	3	3
	TOTAL	64	63	58	58

Scale: 3 = "Yes", 2 = "Not quite or with some workaround", and 1= "No".