Teaching Web Programming
An Approach Rooted in Database Principles

Francesco Maiorana
Department of Electrical, Electronic and Computer Engineering, University of Catania, Viale A. Doria, 6, Catania, Italy

Keywords: Web Programming, Database Driven Applications, Security, Transaction, Stored Procedure.

Abstract: The World Wide Web has become a medium not only for delivering information but also for providing applications that can be used by a variety of instruments, including mobile devices. For these reasons the importance of teaching these skills as well as the difficulties of organizing a vast amount of concepts and technologies in a fast-paced environment driven by strong industrial interest and competition are widely recognized. This work discusses a case study on teaching web programming, presenting an approach suitable for both high school and university courses. The main innovation of the curriculum is the content presentation, which focuses on small incremental steps, spanning the majority of term lab sessions, as well as a teaching a case study which centres on user login management. The case study covers all the major topics in web programming and teaches students to develop, as well as apply and integrate, major database concepts such as transactions, stored procedures and security issues. This lays the foundation for developing a larger project, which is suggested as the final step of the curriculum.

1 INTRODUCTION

According to the National Science Foundation (2012), (Bureau of Labor Statistics), the occupational outlook for Information Security Analysts, Web Developers, and Computer Network Architects between 2010-2020 is expected to increase at a rate of 22%, which is faster than average. The strong interest is confirmed by the increase in web technologies. In (Gundotra, 2009), for example, during his keynote speech at the Google I/O 2009 conference, the Vice President of Engineering at Google stated that, “The web has won. It has become the dominant programming model of our time”.

At the same time, Academia has given web development an increasing level of importance, reflected by the large number of related courses, from entry level to more advanced. Also in research there is a flourishing of conferences and special issue journals such as (Connolly and Miller, 2013) with “two inter-related aims. The first of these is to communicate current approaches to teaching web development. The second is to reignite the conversation about what role, if any, that web development should have in current computing curricula”. The same trend is also evident in K-12 education, where web development is recognized as an important topic in all recently revised curriculum. These include (CSTA 2011), (Computing at Schools, 2012), (New Zealand Ministry of Education 2010), as well as new curricula designed at the national level, such as (Astrachan, 2012) or initiatives developed around innovative ideas on how to teach introductory programming using web technologies such as Code Avengers (http://www.codeavengers.com/) or CodeHS (http://codehs.com/). Web development is also present in undergraduate and graduate curriculum such as (ACM/IEEE, 2013), (ACM/AIS, 2010), (ACM/IEEE, 2008).

From the above cited works, the interest in the subject is evident. However, an analysis of the curricula reveals that it is difficult to cope with all the content and technologies within a single course, especially in high school or in introductory university courses. As proposed by (Connolly, 2011) and (Maiorana, 2013 (a)), the topics should be covered in three courses: basic markup languages and Cascading Style Sheet in the first course; client side programming in the second course; and server side programming in the third course, with usability and interface design issue spanning all three courses. Other research such as (Wang, 2009) points to the

Maiorana F..
Teaching Web Programming - An Approach Rooted in Database Principles.
DOI: 10.5220/0004849300490056
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 49-56
ISBN: 978-989-758-021-5

need for subdividing the web programming curriculum into more courses, suggesting a two-course curriculum.

This work presents a curriculum suited for either a university course on web programming or for a third-level high school course, in accordance with the CSTA curriculum. The curriculum was field tested in a high school at the $13^{th}/13^{th}$ grade with students enrolled in a Computer Science specialization course. The curriculum, by leveraging on prerequisite database knowledge such as Entity Relationship model (E/R) and Data Manipulation language (SQL), focuses on server side web programming and offers the possibility, using simple examples and guided practical lab sessions, to present and practically apply, as well as verify, advanced database concepts such as stored procedures, transactions and security issues, both at the database and at the web application levels.

The paper is organized as follows: section 2 briefly reviews the literature on teaching web development; section 3 presents the context of the teaching experience, the pedagogical model, the contents and how the course was taught; section 4 discusses a first course evaluation, highlighting positive and negative aspects; and section 5 presents the conclusions and highlights future work.

2 STATE OF THE ART IN TEACHING WEB PROGRAMMING

In the research carried out by (Grove, 2007), the authors review ten years of research articles on teaching web development in colleges and universities. The authors report that the diversity of protocols and programming languages required for web development presents a problem. Mastering all the knowledge required is deemed “unreasonable” while some practical level of understanding in all the technologies is considered necessary. This conclusion correlates with the necessity to divide, especially in high school, the curriculum into at least three courses. This is reinforced by (Park, 2013) where a taxonomy of student errors in HTML and CSS is analysed, reinforcing the pedagogical idea that even entry level languages, not considered as “computationally expensive” as many other languages, require developing computational skills such as checking syntax errors and debugging. The work makes use of an educational tool, openHTML, a web editor developed by the authors. Another e-learning system designed for improving web programming skills, is presented in (Elgamal, 2013), and offers an integrated environment for web programming.

Among recent case studies and teaching experiences on web programming at the university level, it is possible to recall the following notable research: (Liu, 2011) presenting challenges and tools used; (Laverty, 2011) where the authors highlight the challenge of an “efficient delivery of a dynamic web development, database-driven platform”; (Baatard, 2007) presenting a course using the PHP language and addressing security issues; (Noonan, 2007) (Wang, 2006) and (Olan, 2009) presenting a course focused on server side programming and database interaction; (Tao, 2010) reporting on the positive effect of encouraging students to use code developed by others; (Hollingsworth, 2010) where the Google Cloud App was used along with servlets and Java Server Pages (JSP); (Gousie, 2006) reporting on an interdisciplinary approach to teaching web programming, graphics and design in a course for non-majors; (Stepp, 2009) presenting a CS 1.5 web programming course, and suggesting that educators present form and PHP scripting as early as possible; and (Adams, 2007) presenting a course developed around a Web project. The use of frameworks, recently reviewed in (Chao, 2013), is not recommended in introductory web programming courses.

From this brief literature review it is possible to highlight the following points:

- the difficulty and challenges in selecting the topics and providing sufficient depth (Liu, 2011)
- the rich set of technologies used, the lack of their integration and their development at a fast pace
- the importance of delivering a dynamic, database-driven web development
- the necessity to span the content of the topics among several courses and the interdisciplinary aspect of the topic.

This work reports a detailed practical experience suited for a web programming module and focuses on how these activities offer the possibility to practically explore and experiment with advanced database concepts such as stored procedure, transaction and security in the realm of web programming. The use of these advanced database concepts in web programming represents a unique case compared to other courses described in literature. The application of the blending of the proposed pedagogical approach, although representing a well-established practice, is presented in a particular context and within a particular discipline and will hopefully provide an initial insight in the described setting.

3 CONTEXT, PEDAGOGICAL MODEL AND CONTENT OF THE EDUCATIONAL EXPERIENCE

This work presents a case study dealing with an educational experience in teaching web programming in an Italian high school. The class comprised 14 students, 12 male and 2 female, between the ages of 18 and 20. The students in Italy typically start school at the age of 6, with a few exceptions who start at 5, and after 13 years of school, finish at 19. Among the students, there were two who had to repeat several years of school. The class was the final course of a computer science curriculum lasting five years. In the last three years the students attended two types of courses each year: the first type of course centers on programming and software development and the second type of course centers on hardware and networking. In the third year programming course, the students have to learn basic programming, problem solving and algorithm design in an imperative language such as C, as well as basic web page design using HTML and CSS. In the fourth year the students learn object- oriented programming along with event driven programming. The students extend the web design topic by writing JavaScript code on the client side. In the fifth year the students in the programming course deal with Database design and SQL language in the first term and web programming in the second term. This content is accompanied by the second course where students learn networking architecture, both hardware and software, client server programming and security issues, such as cryptography. The reported teaching experience deals with the second term of the software course focused on web programming technologies, where the students are required to master database design with E/R diagram and SQL languages both for Data Manipulation and for querying. The course in the fifth year was divided into 28 weeks, 12 of which were dedicated to the web programming curriculum. The final two weeks allowed for individual and group project finalization. Each week consisted of a two-hour session for theory, ground discussions and assessment, and three one-hour lab sessions divided into two hours plus a one hour meeting, for a total of three meetings per week.

The course schedule for the first term is reported in Table 1. For a deeper discussion of the database curriculum the reader can refer to (Giordano, 2013 (c)).

Table 1: Course schedule for the first term.

Unit	Argument	Hours
1	E/R, entity, primary key, attribute	2
	Projection and filtering using one table. Regular expressions	6
	User defined functions	6
	User interface for parametric queries	4
2	Associations 1-N, 1-1; association attribute, multiple associations, association with role.	6
	Join and set operations	8
3	Associations N-N.	6
	Sub-queries. Nested and correlated sub-queries	6
4	Ternary and n-ary associations	8
	Group by and nested queries	6
5	Associative entity, Normalization.	2
	Having clause	4
6	Recursive associations.	6
	Total	70

The bibliography for students comprised an Italian book together with selected parts from two leading database books (Atzeni, 1999) (Elmasri, 2011), as well as teacher notes. The adopted pedagogical model, in accordance with modern constructivist theory (Duffy, 1992), was student-centered, where the students in each lab session are required to design and implement a solution to small problems, presented as specific questions requesting to perform a particular task. The questions guide the students to a final lab session goal of producing an application solving a specific problem or applying a new technology. An example of such questions for the 7th session, as listed in Table 3, is given in Table 2. This approach has the advantage of guiding students in the complex task of mastering different technologies, programming languages, as well as in learning and understanding the deep and fundamental concepts of database and web programming.

According to the revised Bloom taxonomy (Anderson, 2005), the questions guide the students in climbing all the levels of the taxonomy: students using self-directed work along with the lab assignment questions are guided in the cognitive process of remembering, understanding and applying. More gifted students are able to analyze and evaluate, while a guided inquiry approach is used during the lab session to allow the other students to analyze and evaluate their solutions. The creative aspect is saved for the end of the lab, in the form of deeper questions, self-directed exploration and experimentation and, most importantly, in the final project, where it is suggested that the students apply all the course knowledge in building a complete, self-chosen web application. The final project can also be done in self-selected groups of students.

Table 2: An example of a lab session assignment.

N	Questions
1	Create, using the SQL language a database called store.
2	Create, using the SQL language, the following logical model: Person (id, Lastname, Firstname, age, sex, BornDate, salary, StartTime, CivilStatus) Login (id, idP, username, password) where CivilStatus can only have a value chosen from (single, nubile, conjugated, separated)
3	Insert data in the Person and login table. Verify that the reference integrity constraint works.
4	Create a stored procedure to insert a person. Create a function to insert a person returning the ID of the last inserted person. Verify that the stored procedure and stored function work by calling them
5	Create a stored procedure to insert in the login table and verify it. Verify that the reference integrity constraint works.
6	Create an HTML page containing a form to insert the data of a person including username and password.
7	Create a PHP page to insert in the database the data relative to the person including the login information. Verify that the transaction mechanism works by forcing the insertion in the login table of a row with a non-existent value for the foreign key.
8	Write a stored procedure that, by using the transaction, inserts a row in the person and login table, thus inserting all the information pertinent to the person. Simplify the PHP code in order to perform a single call to this stored procedure. Compare the solutions developed in steps 7 and 8.

The lab sessions were designed in an incremental way around the management of the login process and user registration. It was decided, for didactic and security reasons, to divide the user data and the login data (username and password) into two tables allowing the students to present and practically apply important concepts such as stored procedure, transaction, security issues and security advantages in using stored procedure. A modern object-oriented API such as the Portable Database Object (PDO) was used for database access. The login process allowed students to deal with the main aspects of web programming: HTML, forms, PHP, SQL, session and cookies to keep track of the data across the web pages; data validation took place both on the client side using JavaScript and regular expressions, and on the server side using PHP function and database stored procedures. The choice of the model allowed for the application and verification, with few lines of code, of sophisticated mechanisms such as transactions and stored procedure invoker and creator security models, as well as SQL commands for security management. Possible lab extensions to the basic login page could provide forgotten password functionalities, date and time of last login, count of unsuccessful logins on the basis of the username and a personalized welcome message, just to name a few.

Throughout the entire term an inverted classroom approach (Gannod, 2008) was used where the students were asked to pre-read the material, thereby leaving time for practical work during the lab sessions and discussion during the theory sessions. These discussions included solutions to exercises and case studies with critical comparisons of different approaches to solving the problems. The material was based on the textbook as well as the teacher's notes. Modern web 2.0 technologies (www.blogger.com/) (sites.google.com/) were used to deliver content, encouraging students to use wiki technologies to develop a personal e-portfolio where solutions to case studies and projects were posted to allow peers and instructors to leave formative feedback (Giordano, 2004), (Giordano, 2013 (a, b). The free Easy PHP (http://www.easyphp.org/) Windows Apache MYSQL PHP (WAMP) was used on a Windows platform. Table 3 reports the course schedule with the major lab assignment topics.

The lab sessions were very fruitful and allowed for student self-discovery. For example, during a lab session a student working on a machine with an early version of PHP verified that the foreign key constraint does not work with the MYSQL MyISAM engine. It was suggested that the students use the manual to find a way to change the engine and verify that the foreign key constraint is implemented in other engines such as the InnoDB. The side effect was to push students to improve technical manual reading capabilities, particularly for SQL syntax such as DML instructions (MySQL, 2013).

This approach was extended to the PHP manual (PHP, 2013), thus improving a technical skill required in working environments. Overall, the

Table 3: Course schedule and lab assignments.

N	Topic	Lab assignment	#h
1	Three tier architecture	Install and configure a WAMP framework	3
2	HTML review	Login form creation	3
3	Database connection and data insertion through stored procedure.	Stored procedures for insertion in one table and their use with PDO. Parameters. Insert into the person table.	6
4	Stored procedure for updating and deleting data.	Use of HTML forms and stored procedure to update and delete data. Manage the person and login table. Get and POST method.	6
5	Stored procedures for retrieving data with and without parameters.	Retrieving data using stored procedures and PDO and presenting them in HTML table. CSS review. List of registered users.	6
6	Transaction: begin, commit and rollback	PHP code for managing transaction. SQL code for managing transaction. Insertion in the Person and login table	8
7	Use of transaction to insert, update and modify data across multiple tables	Use of transaction to insert, modify data in multiple tables. Management of the person and login table	4
8	Cookies and sessions	Keep track of username and ID across multiple web pages	6
9	SQL command for security	Granting and revoking privileges to users. Stored procedure for granting and revoking privileges	4
10	Security model of stored procedures: definer and invoker model. Cryptography.	Use and comparison of different security models to automatically manage user creation and management. Cryptography functions in PHP.	6
11	Client side form validation using JavaScript and regular expressions	Validate the Person and login data at the client side	4
12	Server side data validation.	Using PHP and stored procedure to validate the person and login data at the server side	4
		Total	60

approach was useful in guiding the students in the complex process of designing web applications, allowing for an incremental project that pushed the students to develop limited but working projects in each lab session, refining the solution throughout the semester. This approach avoided teaching each topic individually but rather fostered step-by-step teaching, through incremental integration.

Initially, the students were reluctant to follow the approach due to the added level of abstraction introduced by the stored procedures and hence the higher cognitive effort required to master them. While proceeding with this approach, their opinions changed positively and the students appreciated the benefit of the method in terms of performance and security. Likewise, they appreciated the reduced burden in writing PHP code and the possibility to resolve the basic database management task with the same approach and by using a similar design procedure.

The teachers' evaluation of the experience, by comparison with previous classes with the same coursework and the same teachers, where more emphasis was given to physical database design, indexing and query performance (Maiorana (b), 2013) thus leaving less time for more extensive web programming, was positive, mainly due to the possibility of exposing the students to a richer set of technology with a tighter link to database technologies and concepts. Overall, 6 students were able to complete all the guided lab exercises independently, with little guidance in the first lab session, 2 students completed the lab session but required more support from the teachers, 6 students struggled due to the difficulty and deficiency both in concepts such as procedure and functions that were covered in the previous years, and to a deficiency in the mastering of some topics related to the first part of the course and finally 1 student was not able to complete the course and pass the year due to the numerous insufficient results in other disciplines. The final student grades in the discipline were: 4 students received an A, 3 students a B, 5 students a C, 1 student a D and 2 students an F.

4 COURSE EVALUATION AND DISCUSSION

The course was evaluated by using student and teacher feedback, obtained by informal discussion and formal colloquia inside the class meetings and by assessing and evaluating student work in each lab session. The feedback from students as well as teachers was positive.

Students were initially overwhelmed by the complication introduced by stored procedure and the necessity to manage and discriminate different working environments such as the MYSQL database, the PHP scripting languages as well as HTML. When the initial difficulties were overcome, thanks to a series of lab sessions focusing on similar concepts and technologies (e.g., stored procedure for inserting, deleting, updating and retrieving data from a single table), the students particularly appreciated using stored procedure and advanced database concepts which allowed them to use less code, perform canonical tasks such as inserting, updating, deleting and retrieving data in a canonical and recurring way, changing only the SQL command to be used in the stored procedure as well as the number and type of parameters.

From our colleagues' point of view, the approach allowed us to present and master advanced database concepts. The approach allowed for the sharpening and augmentation of database knowledge required by all the students in the final nation-wide state examination. The step-by-step approach avoided students getting lost in the multiple environment scenario typical of Web programming, despite the coverage of advanced topics.

The approach allowed students to experiment and easily verify important topics, such as transactions, by verifying their effects. Examples of this verification approach include: inserting a violating reference key data as the second step in a transaction, thereby allowing students to verify the missing effect in the first step; looking for the data stored on the client side (the cookie) and on the server side (the session file), thus allowing students to distinguish between the two concepts and their effects; verifying the different foreign key mechanisms with different MYSQL engines, and consequently mastering this important topic.

The assessment of the lab sessions revealed, through analysis of the results and of the grades assigned, three levels of achievement: the more gifted students were the first to proceed alone and expand the proposed lab session assignments; a second group of students were able to follow the steps and work through a solution to each step with minimal guidance; finally, a third group of students required particular attention and support to work through the exercises, with some of them falling behind and not meeting the deadlines. This division was reflected in the final grade of the module, as noted before. Despite the increased complexity of the course topic, the results correlate with and are slightly better than past experiences.

Overall, compared to a previous course taught by the same teachers, a deeper mastering of the web programming material was observed. Whereas only two students from the previous course were able to apply and discuss stored procedures and transactions in their final project (Maiorana (b), 2013), up to eight were able to do so in this course. This was due to the different teaching approach used, one that did not use the step-by-step assignment method in the lab session and which also dedicated fewer hours to the web programming module in order to explain another important topic, i.e., indexing, which is now an elective course in the new CS curriculum (ACM/IEEE, 2013).

5 CONCLUSIONS

A web programming curriculum was proposed, based on the results of a teaching experience in the final year at an Italian high school. The pedagogical approach was based on lab sessions requiring students to design and implement a solution with a small sequence of steps leading to a working artifact. The lab sessions subsequently refined each other by presenting and applying new concepts or techniques and incrementally improving a project spanning all the lab sessions which centered on the management of the login information on a website. In solving the login problem, the students were able to cover all the major topics of web programming, linking them to advanced database concepts such as transaction, stored procedure and security models, topics not usually covered in a high school or undergraduate setting. At the same time the concepts, mastered by the majority of the students, allowed for an easy framing of a solution pattern for the management of data in Web applications. A preliminary course assessment has also been presented.

The presented curriculum, according to Maiorana (2013 (a)) is suited for a third-level course dealing with server side programming. The course represents the concluding phase, following a tag-based language and Cascading Style Sheet in the first course and client side programming in the second course. The proposed approach presents the advantages of a strong relationship with database concepts, allowing at the same time to present initial issues related to web interfaces and usability.

The proposed curriculum and pedagogical approach can be used in both high school and university courses, allowing the students to present and practically apply all the concepts suited for the third-level curriculum in the proposed three-level course architecture. Regarding further research, the following is planned: a deeper qualitative and quantitative course assessment compared to a previous class with the same coursework and the expansion of the curriculum by using frameworks, such as jQuery Mobile (http://jquerymobile.com/), to develop mobile web applications (Costanzo, 2013).

REFERENCES

Adams, D. R., 2007. Integration early: a new approach to teaching web application development. *Journal of Computing Sciences in Colleges*, Vol. 23, No. 1, pp. 97-104.

Anderson, L. W., Krathwohl, D. R., Bloom, B. S., 2005. *A taxonomy for learning, teaching, and assessing*. Longman.

Association for Computing Machinery (ACM), Association for Information Systems (AIS**).** IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Retrieved on August 2013 from http://www.acm.org/education/ curricula/IS%202010%20ACM%20final.pdf.

Association for Computing Machinery (ACM), IEEE Computer Society. Information Technology 2008. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Retrieved on August 2013 from http://www.acm.org/education/ curricula/IT2008%20Curriculum.pdf.

Astrachan, O., Briggs, A., 2012. The CS principles project, *ACM Inroads*, Vol 3, No. 2, pp. 38-42.

Atzeni, P., Ceri, S., Paraboschi, S., Torlone, R., 1999. *Database Systems: concepts, languages and architectures*. McGraw-Hill, 1999.

Baatard, G., 2007. Teaching PHP with security in mind. In Proceeding of the 5th *Australian Information Security Management Conference*, pp. 21-27.

Bureau of labor statistics. Occupational outlook handbook. Retrieved September 2013 from http://www.bls.gov/ ooh/computer-and-information-technology/ information -security-analysts-web-developers-and-computer-network-architects.htm.

Chao, J., Davey, B., 2013. Navigating the Framework Jungle for Teaching Web Application Development. *Issues in Informing Science and Information Technology*, 10, pp. 95-109.

Connolly, R., 2011. Awakening Rip Van Winkle: modernizing the Computer Science Web Curriculum. In Proceedings of the 1*6th annual joint conference on Innovation and technology in computer science education (ITICSE, 2011)*, pp. 18-22, ACM, 2011.

Connolly, R. W., Miller, C. S., 2013. Teaching Web Development. Special Issue of the *ACM Journal Transactions on Computing Education*. Retrieved on August 2013 from http://toce.acm.org/attachments/ OCE-SE-TW.pdf.

Costanzo, A., Faro, A., Giordano, D., 2013. WI-CITY: living, deciding and planning using mobiles in Intelligent Cities. In *3rd International Conference on Pervasive and Embedded Computing and Communication Systems, PECCS, Barcelona, INSTICC*.

CSTA Standards Task Force. 2011. K–12 Computer Science Standards Revised 2011. The College Board, Computer Science: Principles Big Ideas and Key Concepts Learning Objectives and Evidence Statements.

Duffy, T. M., Jonassen, D. H., (Eds.), 1992. *Constructivism and the technology of instruction: A conversation*. Psychology Press.

Elgamal, A. F., Abas, H. A., & Baladoh, E. S., 2013. An interactive e-learning system for improving web programming skills. *Education and Information Technologies*, *18*(1), pp. 29-46.

Elmasri, R. & Navathe, S., 2011. *Fundamentals of Database Systems* (6th Edition). Addison Wesley.

Giordano, D., 2004. Shared values as anchors of a learning community: A case study in information systems design. *Journal of Educational Media*, 29, 3, pp. 213-227.

Giordano, D., Maiorana, F., 2013 (a). An Interdisciplinary Project in Sustainable Development Based on Modern Visual Programming Environments and Web 2.0 Technologies. Proceedings of the *3rd Interdisciplinary Engineering Design Education Conference, IEDEC* 2013, pp. 163-167.

Giordano, D., Maiorana, F., 2013 (b). Giordano, D., Maiorana, F., 2013. Object Oriented Design through Game Development in XNA. Proceedings of the *3rd Interdisciplinary Engineering Design Education Conference, IEDEC* 2013, pp. 51-55.

Giordano, D., Maiorana, F., 2013 (c). Teaching database: a pedagogical and curriculum perspective. Proceedings of the *International Conference on Information and Communication Technology for Education, (ICTE)*, in press, 2013.

Gousie, M. B., 2006. A robust web programming and graphics course for non-majors. *ACM SIGCSE Bulletin*, Vol. *38*, No. 1, pp. 72-76.

Grove, R. F., 2007. Trends in teaching web-based development - A survey of pedagogy in web development courses. Proc. 3rd *International Conference on Web Information Systems and Technologies, Webist*, Barcelona, Spain.

Gundotra, V., 2009. Google I/O 2009 Keynote Day 1. http://www.youtube.com/watch?v=S5aGZIvk&feature =channel.

Gannod, G. C., Burge, J. E., Helmick, M. T., 2008. Using the inverted classroom to teach software engineering. In Proceedings of the *30th international conference on Software engineering*, pp. 777-786, ACM.

Hollingsworth, J., Powell, D. J., 2010. Teaching web programming using the Google Cloud. In Proceedings of the *48th Annual Southeast Regional Conference*, pp. 76-80, ACM.

Laverty, J. P., 2011. Implementing a dynamic database-driven course using LAMP. *Information System Education Journal (ISEDJ)*, Vol. 9, No. 7, pp. 33-40.

Liu, Y., Phelps, G., 2011. Challenges and professional tools used when teaching web programming. J*ournal* *of Computing Sciences in Colleges*, Vol. *26*, No. 5, pp. 116-121.

Maiorana, F., 2013 (a). Teaching web programming: critical considerations on pedagogical approach and curriculum content, Proc. of International Conference on Sport Science and Computer Science.

Maiorana, F., Giordano, D., 2013 (b). A constructivist approach to teaching index selection strategies and database design. Proceedings of the *International Conference on Information and Communication Technology for Education, (ICTE)*, in press, 2013.

MySQL 5.6 Reference Manual. Retrieved August 2013 from http://downloads.mysql.com/docs/refman-5.6-en.a4.pdf.

National Science Foundation (US) National Science Board. 2012. Science and Engineering Indicators.

New Zealand Ministry of Education, 2010. Digital technology guideline, dtg.tki.org.nz.

Noonan, R. E., 2007. A course in web programming. J*ournal of Computing Sciences in Colleges*, Vol. 22, No. 3, pp. 23-28.

Olan, M., 2009. Web applications: a test bed for advanced topics. *Journal of Computing Sciences in Colleges*, Vol. 24, No. 3, pp. 72-80.

Park, T. H., Saxena, A., Jagannath, S., Wiedenbeck, S., Forte, A., 2013. Towards a taxonomy of errors in HTML and CSS. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 75-82). ACM.

PHP manual. Retrieved August 2013 from http://us3.php.net/get/php_manual_en.html.gz/from/a/mirror.

Stepp, M., Miller, J., Kirst, V., 2009. A CS 1.5 introduction to web programming. In *ACM SIGCSE Bulletin* Vol. 41, No. 1, pp. 121-125, ACM.

Tao, Y., Plouff, C., 2010. Work in progress: An opportunistic approach for learning web application development. In *Frontiers in Education Conference (FIE)*, 2010 IEEE (pp. S2D-1), IEEE.

The Computing at School Working Group. Computer Science: A curriculum for schools (2012). Retrieved August 2013 from http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf.

The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society. Computer Science Curricula 2013. Ironman Draft. Retrieved Agust 2013 from http://ai.stanford.edu/users/zsahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf.

Wang, X., 2006. A practical way to teach web programming in computer science. *Journal of Computing Sciences in Colleges*, Vol. 22, No. 1, pp. 211-220.

Wang, Y. D., Zahadat, N., 2009. Teaching web development in the Web 2.0 Era. In Proceedings of the *10th ACM SIGITE conference on Information technology education*, pp. 80-86.