WIDE: Centralized and Collaborative Approach to Teaching Web Development

Aleksandar Jevremovic[1], Nenad Ristic[2], Mladen Veinovic[1], Goran Shimic[3], Natasa Stanisic[4]

[1] Informatics and Computing Department, Singidunum University, Serbia
[2] Informatics and Computing Department, Sinergia University, Bosnia and Herzegovina
[3] Military Academy, University of Defense, Serbia
[4] Singidunum University, Serbia
ajevremovic@singidunum.ac.rs, nristic@sinergija.edu.ba, mveinovic@singidunum.ac.rs, gshimic@gmail.com, natasastanisic@singidunum.ac.rs

Abstract

Teaching Web development is an increasingly important and complex task due to multiple technologies that students have to master and implement in each particular solution. This fact imposes the specific learning approach and development environment as a collaborative learning tool. It is particularly important in the learning scenarios that include large groups of students and computers that are not intended solely for teaching Web development as well. The authors of this paper find, based on their past experiences, that a disproportionately large amount of time is expended on troubleshooting infrastructure problems, and that collaboration amongst students is unsatisfactory. This paper presents a solution for centralized and collaborative work on learning Web development, as well as observations made during the course of its development and the first year of deployment.

Keywords: e Web development, On-line IDE, Education, Collaboration

1 Introduction

Teaching Web development is an increasingly important and complex task. Web represents the main infrastructure for the most of the contemporary information systems. This infrastructure consists of numerous of technologies specialized for particular purpose. They have to be used together in each particular (business or non-business) solution. Their permanent changing makes the complexity of their learning more difficult. Moreover, the individual learning on this matter is not getting satisfactory results due the time necessary for students to master these technologies. Fact that students should be able to simultaneously understand and integrate multiple technologies imposes the necessity of converting the learning environment.

The development environment issue is particularly important when working with larger groups of students and on computers that are not intended solely for teaching Web development. Individual monitoring and error-correction of students have proved to be very time-consuming tasks for instructors, and are also sub-optimal with regards to the use of the students' time. In addition, the final result is often a development environment that contains residual errors in component configuration, which will be made evident in the coming classes.

This paper presents a collaborative tool built on Linux platform using open-source Web technologies designed for learning Web development that enabling the teachers to supervise the progress of each particular student during their team work on web application. In order to use the environment, the students need to have only a web browser installed on their computer.

Related works are presented in the next section; proposed solution is presented in the third one. The collected experience in solving several unexpected problems, specificities and options as well as achieved results are presented in the following section. Finally, the conclusions are presented in the last one.

2 Related Work

Education in conjunction with Web development is a topic that has been under active scrutiny for nearly two decades and is still very relevant today. The results of a number of studies were used in the development of the WIDE system, of which the following deserve special mention.

Lim published a paper about teaching Web development technologies in CS/IS curricula in 1998 [1]. The same year, Mercuri et al. published a paper on the use of HTML and JavaScript in introductory programming courses in 1998 [2]. Next year Walker

*Corresponding Author: Aleksandar Jevremovic; E-mail: ajevremovic@singidunum.ac.rs
DOI: 10.3966/160792642018081904004

and Browne published a paper about teaching Web development with limited resources (limited time, diverse student population, and limited laboratory facilities) [3]. In 2000 Klassner dealt with the problem of obsolescence of Web development courses [4]. He proposeds a conceptual framework for a course on Web development titled Internet Application Design: Theory and Practice.

Next year Murugesan et al. published a paper that gives an introductory overview of Web Engineering [5]. It presents the principles and roles of Web Engineering, assesses the similarities and differences between development of traditional software and Web-based systems, and identifies key Web engineering activities.

In 2002, Haney published a paper [6] with lessons learned from incorporating an XML component into an upper division web development course. By experiencing the step-by-step process of the two options (with use of XML or without it), the students learn the costs and benefits, as well as the strengths and weaknesses of both.

In 2004, Yue and Ding presented a paper [7] about the design and evolution of an undergraduate course on web application development. They recognized a necessity the edge Web technologies should be incorporated in a course as soon as they become mature and stable. Also, they highlighted importance of in-depth study of a selected technology to provide the necessary depth and knowledge to build realistic Web applications.

In 2006, Ellis published an evaluation of learning in an online project-based web application design and development course [8]. This paper compares the performance of students on a team project in online and face-to-face sections of a project-based Web Application Design and Development course. Performance factors examined include the number of iterations and the number of days required to finalize project documents, course grades, project grades, and results of a survey seeking student input on the project experience and the overall opinion of the course.

In 2006, Gribble published a paper on the use of Macromedia Breeze in an advanced web development course delivered face-to-face and distance students [9].

In the same year Wang wrote about a practical way to teach web programming in computer science [10]. This paper discusses the details of the design and implementation for the web programming course, including course contents, structure, implementation, demonstration, assignments, final project, outcomes and feedback.

In 2007, Robert Adams published a paper about a new approach to teaching web application development, based on early integration [11]. In this paper the author describes a novel approach to teaching web application development whose focus is on integrating web topics early and often.

In 2009, Wang and Zahadat published a paper about teaching web development in the web 2.0 era [12]. This paper presents the authors' shared experiences and knowledge in teaching and training students in developing Web 2.0 websites in a sequence of two high-level Web development courses.

In 2010, Kazmerik wrote [13] about student's perspective on the Web development courses the effectiveness and how it impacted students' maturation as a developer of web technologies.

In 2011, Al-Khalifa and Al-Rajebah published a paper about integrating mobile web development into IT curriculum of an advanced web technologies course focused on students' feedbacks [14]. In the same year, Park and Wiedenbeck wrote about challenges at an earlier stage of computing education while learning web development [15]. The authors applied content analysis on help-seeking activity of forty-nine students in an introductory web development course. Help forum was used as a tool for this purpose and posted content was analyzed in the research. The authors apply a second level of content analysis to the development issues, identifying aspects of learning HTML, CSS, and JavaScript that challenged students most often. Finally, they identify several computational concepts that relate to these challenges, including notation, hierarchies and paths, nesting, parameters and arguments, and decomposition and abstraction.

In 2011, Connolly published a paper about modernizing the curriculum of computer Web science [16]. The paper critiques the approach of teaching web development topics within a single course. The paper concludes that a multi-course stream in web development can help the students integrate the discrete pieces of knowledge garnered during their undergraduate education.

In the same year, Wang published a paper about teaching web development at a distance [17]. This paper aims to fill the gap of lacking literature by providing instructional details of teaching a Web development course in the format of asynchronous distance education and offering practical instructional strategies. This paper described comparative example of an advanced technical course that has been taught traditionally (residentially, face-to-face), and on distance in the same time (by using problem-based learning methods).

In 2012, Gupta et al. published a paper about utilizing ASP.NET MVC in web development courses [18]. This paper provides an overview of ASP.NET MVC and briefly explains web development using this technology. The paper concludes with examples of course integration and a summary.

In the same year, Hamid published a paper about automated web-based tool which produces interactive user interfaces adapted to for novice programmers [19]. Novice programmers can thus develop programs focusing on design, implementation, and testing of their algorithm (realized as a function of appropriate parameters) and ignore the intricacies of handling user I/O. The web interface is more impressive than console I/O and more simple to assemble than a traditional GUI, requiring no knowledge of web technologies to setup and launch.

Further, Mudigonda and Buerck published a paper about design of an undergraduate web application development course using free technologies and learning materials [20]. The paper discusses some of the constraints imposed by the short duration of the course, and describes the criteria used in the selection of topics, learning materials, class activities and evaluation methods. While the experience described here is based on a week term, it can serve as a basis for the design and implementation of a course in web-application development that is offered terms of other durations as well.

In the same year, Miller published a paper about metonymic errors in a web development course [21]. This paper investigates a class of database access errors that occur in the context of a web development course. While the use of an Object-Relational Mapping (ORM) simplifies database access, students still demonstrate reference errors such as mistakenly referring to the whole object rather than an attribute value that is a part of the object. Metonymy, a rhetorical device used in human communication, offers an interpretation to these errors. A study is presented where student answers are reported and analyzed in this context. Findings indicate the prevalence of reference errors and offer instructional strategies for addressing them.

In 2013, Wang and McKim wrote about teaching web programming in the computer science curriculum CS2013 [22]. They proposed two teaching models they designed as examples for other instructors to use them as references in developing and teaching similar courses in the CS curriculum.

In the same year, Lancor and Katha analyzed PHP frameworks for use in a project-based software engineering course [23]. The top six PHP frameworks (Zend, Yii, CakePHP, CodeIgniter, PRADO, and Symphony) were initially considered and then narrowed down to two (CakePHP and CodeIgniter) based on their alignment with common functionality in previous class projects, framework complexity for those new to frameworks (learning curve), and developer friendliness (availability of documentation and online resources). An in-depth comparative study is conducted by developing a functionally-equivalent web application using each of the two frameworks, as well as plain PHP (no framework).

Further, Dugan presented a model for a single semester web programming course for second year students that they have refined over five years [24]. The course focuses on building both of software engineering and technological skills. A key differentiator between their course model and other models reported in the literature is their use of the Java-based Wicket web programming framework. Wicket provides an intuitive mapping between pure HTML and pure Java and reinforces the software engineering skills taught in the course.

In 2014, Liu gave a talk about streamlining courses for the web development concentration of an IT bachelor degree program [25] that is structured to cover the most important aspects of web design and web development.

In the same year, Hoar explained how institutional IT affects the delivery of a capstone web development course [26]. On the one hand the professional Web developer should be able to configure many different services, while on the other the institutional IT policies preclude working just with some of them

In 2015, Frees published a paper about using node.js in the computer science curriculum [27]. The platform described in the paper supports a smooth learning curve by allowing students to build knowledge gradually through the use of modular, open source components. This paper details the advantages of using node.js in an undergraduate web development course and examines its value to the CS curriculum as a whole.

3 Previous Experience on Web Development Courses

Related works mentioned in the previous section impose natural conclusion that Web Development e-learning strongly depends on contemporary technologies. It should support these technologies by the appropriate e-learning content, pedagogy and methodology in the same time. From the pure technical prospective, Web development e-learning tools have been evolved from time of dedicated computers, through the period of multipurpose computers, which have been replacing with cloud solutions based on virtual machines and sharable Web servers. Next are the pros and contras considerations on this matter.

Dedicated computers, which are used by only one group of students throughout the course, remain the best solution with respect to performance, reliability and freedom of adjustment. A major advantage of this approach is the possibility of providing administrator accounts/privileges to students, which also allows for the possibility of going through the process of installing Web development and production components independently. On the other hand, when students self-install these components, errors may occur, which need to be identified and eliminated by instructors during the course, and each course cycle requires a complete reinstallation of the software. Perhaps the biggest drawback of this solution is its high cost, arising from the purchase of computers and providing classrooms in which they are to be located.

Multipurpose computers represent one of the most common approaches to supporting IT courses. That means the computers that are used for several different courses at the moment (e.g. during one semester). The biggest advantage of this approach is that it saves the cost of having to purchase a computer, as well as allowing the use of one classroom for several courses. However, this approach raises a number of issues. The most common problem is that the students have limited privileges, usually without opportunity to pass through installation and configuration processes important for Web development. It influences directly in quality of education. In addition, computers timely became overloaded with a large amount of installed software and data. This often scenario causes slowdown of operational abilities as well as a conflict of certain software components. Such characteristic could produce vulnerability of Web development courses.

Virtual machines enable the complete software and content isolation on the same hardware platform. That supports two important scenarios: the first one is the computer can run different courses on different virtual machines in the same time and, as the second one, it is easy to set up all computers in the classroom in the same way – same virtual machine. This is very flexible and efficient approach. Instructors can create virtual machine templates and copy them onto student computers. Using this approach, it is possible to provide administrator privileges (on the virtual machine) to students without risking any damage to the software installed on the physical machine. In the event of accidental damage to the virtual machines on student computers, they can easily be replaced with a copy of the prepared template. Only damage is a loss of student files.

There are several disadvantages of such an approach. Physical computers should have high performances for the smooth execution of the virtual machines - fast CPU, large operating memory as well as huge disk space. Moreover, one should not underestimate the burden placed on the local network and the time required to make the initial copy of the virtual machines on student computers because their size is typically several GB. Accordingly, the prices of physical computers with the required characteristics typically exceed the funds allocated for the purchase and upgrading of computers in student laboratories, which is the main reason why this approach is infrequently employed.

Shared Web servers generally represent appropriate solution for courses about Web development. In this approach, a dedicated server with Web hosting components (HTTP and DB Services), as well as components for uploading files to the server (FTP or SSH), is placed on a local area network. After installation and server setup, student accounts are created on the server. Installation of this server eliminates the need to have production components installed on student computers; instead, they only need to have an IDE, database tools, client software for file transfer and a Web browser.

Excluding HTTP and DBMS services reduces the student computers load. Based on previous experience [28] a single server with an average capacity, running the Linux operating system, with an installed and properly configured Apache/PHP Web server and MySQL DBMS can support over 100 students working in parallel without issue.

Using this approach, students are also trained to use the FTP/SSH protocol to upload files onto a server, which is a common practice in the real-world environment. An additional advantage in comparison to the above-mentioned approaches is the students' ability to utilize different computers for different exercises, because they can download their previously developed scripts from the server.

4 WIDE Framework

The WIDE framework is basically client-server web application. On the server side, there is HTTP server with PHP programming language support. In addition, a relational DBMS represents the backend of the server side. Framework is deliverable through different technologies. The most cost effective solution advised by the authors is Linux, Apache and MySQL as a server back bone solution. Only condition for alternative solutions is that they should have PHP programming language support. Contemporary IDE offers this kind of support (e.g. Visual Studio, Eclipse, NetBeans, etc.). On the backend side WIDE framework employs several databases on the server. They can be divided into two categories. WIDE system management databases and project databases. There is just one database of the first type. Its purpose is to store and manage information about existing projects and to track user activities. The other database type is designed for students' projects. For instance, when individual student or team creates new project, the system creates appropriate database for storing project's information as well as other entities and data entered during project development (e.g. configuration files, resources used in the projects, scripts, source code of classes and user defined functions, etc.). By default, each project creates a single database (which bears the same name as the project itself), but, if needed, several databases may be created for a single project.

4.1 Basic Concepts and Principles

Below is conceptual model of WIDE framework. Project represents its main concept (Figure 1). Initially, teacher delivers the concrete task to the students through the project form. In this very beginning phase teacher creates a project task making its description, enrolling the students in it and enabling them access to the necessary resources (such as Web server file system, functional and class libraries, and newly created database). Thereafter teacher can track the changes in the project, see students posts and communication and give them instructions or help on demand. Two application modules are developed for this purpose: WIDE (Web Integrated Development Environment) delivered to the students through the Web browser and Project Manager application for the teachers.

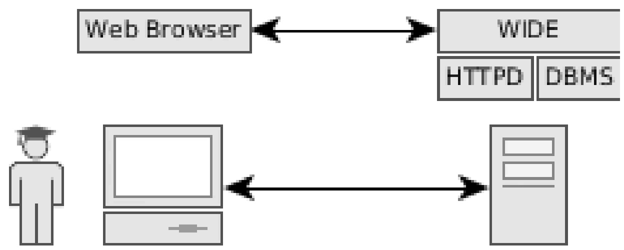

Figure 1. The client-server architecture of the solution

The client side of the application was built using standard web technologies – HTML, JavaScript and CSS. The Web browser is the only component that must be present on the client computers, and, at the same time, this requirement is one of the main motives for the development of this solution. Operation in this environment has been successfully tested with Internet Explorer, Mozilla Firefox, Google Chrome, Opera and Safari browsers, but compatibility with Firefox and Chrome browsers has been prioritized as students seem to prefer them.

Client- and server-side communication is performed using the standard HTTP protocol. The initial loading of an IDE and its components is accomplished using standard GET requests while the subsequent communication with the server is done using AJAX calls. These subsequent calls are logged, which makes it possible to reconstruct student progress at a later time using these records.

The two main components of the WIDE solution are the student application and the instructor application. The student application comprises an Integrated Development Environment, i.e. a Web application that could enable project development. The instructor application is named Project Manager due to the fact that its role is to manage student projects.

As mentioned, each project includes files of different types and separate database. During the development these resources are managed from both the student application and the instructor one (Figure 2).

Since all projects and their resources are located on server side, they are accessible for students and instructors in any time and any place. Namely, a student can use any computer with Internet connection and continue to work on the project started earlier. This advantage can be hardly provided with some other approach (see previous section).

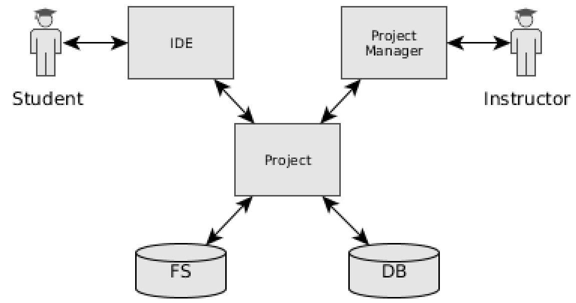

Figure 2. Components and environment organization

The students' collaboration on projects represents an important feature of the WIDE framework. Owing to Web technology implemented, students can access the same project and to work simultaneously on various project files. In practice, this has proved beneficial when matching students of different profiles, especially those who prefer to focus on design and those who prefer to develop the server side of the project. However, perhaps the main advantage of collaboration among students is solving the issue of orchestration, i.e. harmonization of the rate of student advancement.

Collaboration among students is recognized as a major potential of the solution. WIDE framework facilitate the teachers to be much more engaged in a team work of their student than before. On the other hand, work of particular student is more transparent that enables better measurement of the results. In addition, being based on Web technologies and the ability to connect to the Internet open up new possibilities in terms of asynchronous collaboration and more advanced approaches [28].

4.2 Student Application

Student application (WIDE) represents one of two separate modules delivered through the WIDE framework. As mentioned, it represents integrated development environment delivered by using Web technologies (Figure 3). The following functionalities are supported in WIDE: file management at the project level, editing of file content and executing the project or individual scripts. The WIDE components are loaded and executed within a Web browser (client) while the project itself is stored and executed on the server. Such an organization requires minimal resources on the client side. Other words, computers with very limited performances can be used for serious Web development.

WIDE is organized similar as any other IDE. Project is represented as a folder that consists of different kind of resources: built in functional and class libraries

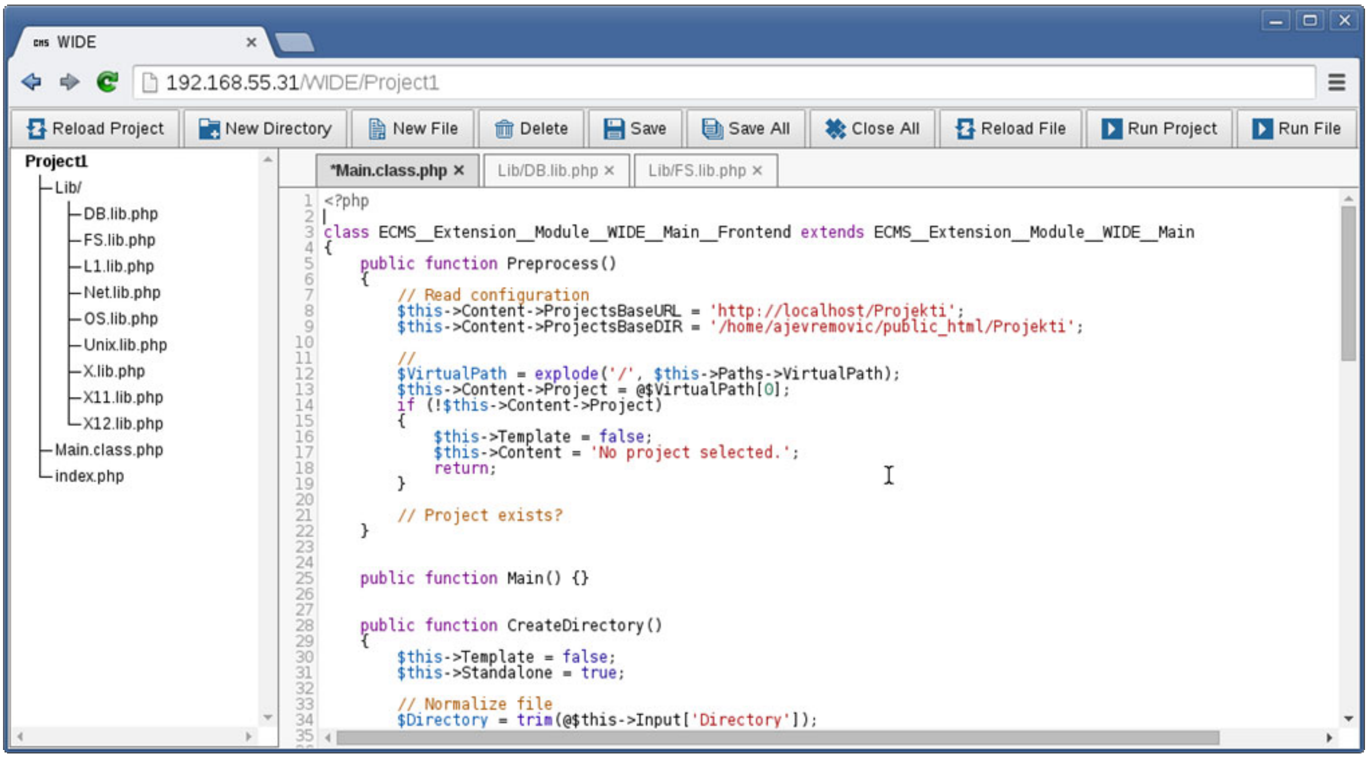

Figure 3. A classical IDE interface replicated in Web technologies

included in project by default or by the teacher, source code files created by the students as well as resource files uploaded with some specific project purpose (e.g. multimedia files, archives, etc.). For text editing functionality, the ready-made open-source component called CodeMirror [30] was used, which is publicly available under the MIT license. CodeMirror is a versatile text editor implemented in JavaScript for the browser. It is specialized for editing code and comes with a number of language modes and add-ons that implement more advanced editing functionality. The significant features of this text editor include auto-completion [29], code folding, syntax highlighting, bracket and tag matching, etc. As student saves a file he worked on, these changes become visible for others enrolled in the same project. Versioning mechanism is used for this purpose. Other words, each file is annotated to preserve a current state of source code in particular moment of time. This characteristic enables full tracking ability to the teachers. The can read the programming code entered by each student and evaluate his engagement on overall project in more proper way.

Collaborative editing of the same file represents the complex problem. We considered a number of implementations. The most of them are based on RESTful Web services and rich client JavaScript that provide push and pull messaging between clients and server, enabling the students 'near real time' collaborative editing.

The solution implemented in a WIDE framework is the original one. In collaborative mode server application creates as clones of the same file as there are students in collaboration. This way individual student edits his own file which can also be changed by the server application. A module based on 'Observer' design pattern that enables the server if the content is changed in any of replicas, all others will be updated in the same way.

Potential collision could be happened if two or more students edit the text at the same position in the file(s). It is solved by putting this content in so called 'History thread'. This way beside the content created on his own, the student can see the work of the others on the side panel. He can use this content to improve his own code. On the other hand the system holds the information about the origins (authors) of each chunk of code. This data is used for after activity analysis of engagement of each individual in joint collaborative work.

4.3 Instructor Application

As mentioned, the instructor application (Project Manager) allows the creation of new projects, as well as modifying, re-creating and removing existing ones (Figure 4). When creating projects, an instructor specifies its name, incorporate a specific set of files into the project and define a list of SQL commands that can be executed during a project development. For instance, teacher can enable students to create / modify / drop tables, to insert / update / delete data in / from tables, to create views, queries and stored procedures. On the other hand, WIDE framework automatically create project database fulfilled with the data and file system with required resources. Such behavior makes two important advantages: first one is that WIDE enables inexperienced students to be quickly engaged into the Web development projects; dealing with advanced projects with improved efficiency provided by using already prepared databases and project setup represents another one.

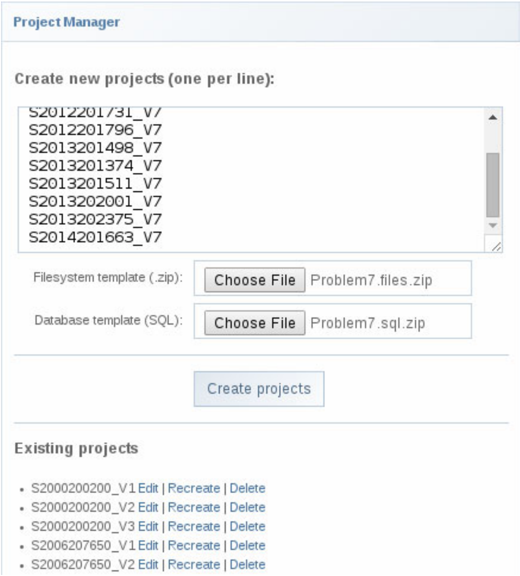

Figure 4. Instructor application (Project Manager)

In the very beginning phase students have difficulties in basic things. Very often they make mistakes in naming policy. Sometimes they lose themselves in layered and modular architecture which is typical for Web development, or have problems in dealing with data exchange between application and database. There are several limitations to naming projects: the maximum length is sixteen characters, Latin letters, numbers and underscore are allowed and the name cannot start with a number. In principle, these restrictions reflect the limitations inherent to naming MySQL databases since the name of the database on the server is followed by a project name.

Beside creating new projects, the instructor application may be used for managing existing projects. For example, each of the existing projects can be edited, which means opening the IDE of the application used by students. In addition, a project can be re-created or restored to its initial state. This option is useful in cases where students fail to identify the path to the solution and need to be returned to the starting position.

Finally, if needed, the project can be deleted entirely. However, taking into account the low volume that the projects take up, this option has rarely been used. In addition, we believe that the code created by students, whether correct or not, is nonetheless valuable for future analyses.

4.4 Cross Referencing

WIDE framework represents distributed system. By default, it requires three IP addresses. The first one is used by students to access the development environment. The second is used as a DB (database) server IP address and the third one is dedicated for running the projects. DB server address is used by students to connect data sources to their projects and to manage it. Separation of the third address is particularly important, primarily in order to ensure that the running project has a clean session, as well as to obtain an isolated domain for cookies and Web storage. The use of cookies and Web storage may pose a significant problem, especially in the case of projects that focus on them specifically, when the same client computer is used by multiple users (i.e. Web browsers are by default setup to collect history data in their internal persistent memory).

In practice, an ideal scenario would be that a new IP address is assigned to each project. However, such a possibility is unavailable to us in practice. It also proved useful to introduce students to the options of removing data history in the browser, as well as to the “incognito” mode of operation of modern browsers.

For all three mentioned IP addresses, separate servers should be used. Minimalistic approach would be deployment of entire WIDE framework using virtual machines on a single server with three separate network interfaces.

4.5 Integration with Local CMS and LMS

Although the WIDE framework can act as a standalone solution, in practice, it is connected to the local Learning Management System (LMS) and University IS (Figure 5). The reason is that the students should access the services through the LMS (as any other e-learning activity and resource). This way students are fully authenticated and verified that they can access to such specific resource (e.g. if they are enrolled for a course and if they have fulfilled demanded preconditions). However, these two systems are low coupled. Links are used just for exchanging between project executive address and the database server. For validation of student data, the built-in feature of an LMS and IS Web service is used.

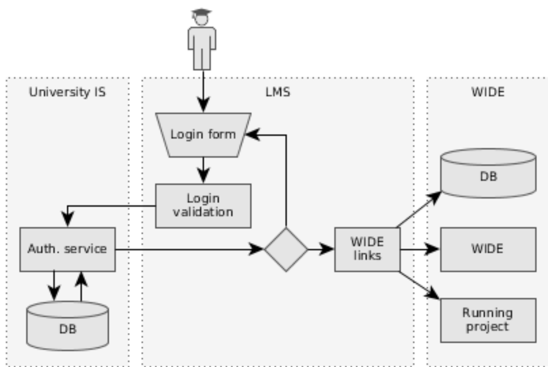

Figure 5. Integration architecture

Upon successful login, a default page is shown to the student with a link to the development environment, a link to the executive variant of the project and the parameters for accessing the project database.

The home page is dynamically composed. Also it can be customized by the teacher in accordance with each individual project. For each project, its unique name, i.e. the address used to access the project, as well as the parameters for accessing the database, need to be changed.

In addition, the teacher has the ability to enter or link the text of an assignment (project requirements), additional materials and so on. To change these messages, the teacher is expected to be conversant with HTML and the Smarty template engine. However, given the area that these courses cover, the teachers are assumed to be fully competent in both.

5 Evaluation and Remarks

The presented environment is in use at several institutions of higher education where the authors work. In this part of the paper, the results of its usage will be presented in terms of the impact on student performance in different school years. In addition, the paper shall present and elaborate on the authors' findings and remarks that may clarify some of the decisions related to the design and development of the solution and could also serve as a useful source for other authors exploring this issue. WIDE solution was developed as a client-server web application. On the server side, the use of an HTTP server with support for the PHP programming language (in which the server part of the solution was written) is required. In addition, a relational DBMS is required on the server. In practice, the authors employed Linux, Apache and MySQL software for the server basis, but compatibility with alternative solutions (compatible with PHP) is expected.

5.1 The Impact on Student Performance

In order to evaluate the impact of the proposed solution on the efficiency of the teaching process, we compared the performance of two generations of students. The first generation, in the academic year 2013/2014, used a traditional environment for work, shared desktop computers, and, in this analysis, these students constituted the control group. The next generation of students, in the academic year 2014/2015, used the proposed environment, and, in this analysis, these students constituted the experimental group. In the first generation, the course was taken by eighty-six students, while in the generation 2014/2015 this number amounted to ninety-one. Student performance was analysed from two aspects: the total number of laboratory exercises and the number of points achieved on the final examination.

The first aspect was the number of laboratory exercises that students managed to perform successfully. This course consisted of eight mandatory laboratory exercises and four additional exercises. All exercises involved the creation of a Web application with a certain level of complexity. The task in the eighth exercise was to develop a Web application using several related database tables. It also required making use of CSS and JavaScript technologies. The four additional exercises were intended for students interested in the AJAX approach, performance issues in database operation, use of the Memcached system and cryptographic functions.

Figure 6 shows a diagram with the percentage of students who managed to complete the specified number of exercises (in total). The diagram shows a reduced dropout rate for the class of 2014 after exercises 3 and 4, and a significantly higher number of students who completed the four additional exercises successfully. We believe that the better performance of this generation is due to the use of the proposed solution, primarily with regard to collaboration, i.e. the assistance provided by students who have completed the exercise successfully to those who have not yet succeeded.

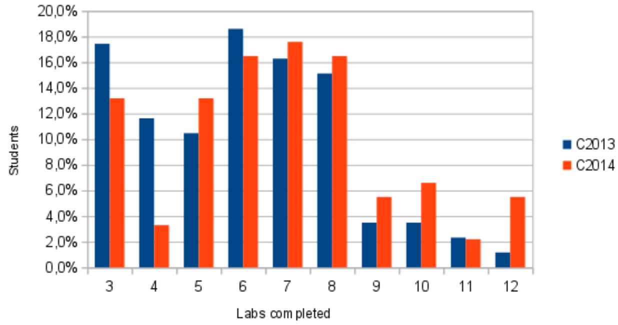

Figure 6. Diagram with the final number of successfully performed laboratory exercises

A statistical analysis of final student grades was conducted using a T-test (Table 1). The student class of 2013/14 was selected as the control group while the class of 2014/15 (the class that had used the system) was selected as the experimental group.

Table 1. Statistical analysis of differences in the total number of completed exercises

Intermediate values used in calculation			
Control Group (2013/14)		Experimental Group (2014/15)	
Mean	6.03	Mean	6.78
SD	2.20	SD	2.41
SEM	0.24	SEM	0.25
N	86	N	91
Standard error of difference			348
Degrees of freedom			175
T value			2.1447
Confidence interval			
CG mean - EG mean			-0.75
Confidence interval (95%)			-1.43 to -0.06
P-value and statistical significance			
P-value (statistically significant)			0.0334

The mean and the standard deviation of the control group were 6.03 and 2.20 respectively. With the experimental group, the mean was 6.78, and the standard deviation was 2.41. The comparison of the results with the corresponding values in the T-table yielded a statistically significant p-value of 0.0334 (calculated on the basis of degrees of freedom for both groups), indicating a statistically significant difference between the results of the two groups being compared.

Another aspect is the total number of credits achieved by students in this course. The diagram in Figure 7 shows that, among the students who obtained credits for the highest grade, there was a significantly higher number of students using the presented solution. Furthermore, no student in this generation achieved a result below 40 percent, and only five students achieved less than 50 percent of credits (compared to nine students in the previous generation).

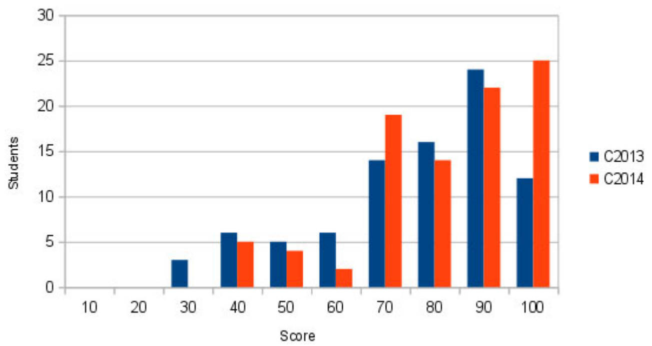

Figure 7. Diagram of the total number of points achieved during the course

T-test analysis of the results also showed that the generation who used the presented solution had a statistically significant better performance (Table 2).

Table 2. Statistical analysis of differences in the overall course performance

Intermediate values used in calculation			
Control Group (2013/14)		Experimental Group (2014/15)	
Mean	72.23	Mean	77.54
SD	18.49	SD	17.22
SEM	1.99	SEM	1.81
N	86	N	91
Standard error of difference			2.685
Degrees of freedom			175
T value			1.9764
Confidence interval			
CG mean - EG mean			-5.31
Confidence interval (95%)			-10.60 to -0.01
P-value and statistical significance			
P-value (statistically significant)			0.0497

5.2 Questionnaire

A questionnaire was created for students for whose classes the proposed solution had been utilised with the aim of obtaining additional information. Students assessed the solution on a scale of 1 to 5 across the following criteria: performance, usability for real-world projects, ease of use and collaboration. The results of the survey are presented in Table 3.

Table 3. Questionnarie results

	1	2	3	4	5
Performance	1	4	58	23	5
Usability	8	17	55	8	3
Easy of use	3	7	38	29	14
Collaboration	3	0	37	20	31

The results indicate that the majority of students have a neutral view on the environment across all criteria. Students are slightly more satisfied with the environment's performance and ease of use, and awarded the highest marks to the environment's collaborative capabilities. On the other hand, a significant number of students consider the environment ill-suited to real-world projects.

6 Conclusion and Proposals for Future Work

In this paper, a centralized solution for Web development, as part of the teaching process, is presented. The solution was implemented as a client-server web application using open-source technologies. The application of this solution has greatly facilitated setting up working environments for students doing Web development assignments. Moreover, through the use of this solution, instructor interventions in student projects have been facilitated, and student collaboration has been simplified.

The effectiveness of this solution was evaluated by comparing the performance of two generations of students, with the former using a traditional environment, and the latter using the presented solution in their work. The impact of the solution was measured by the number of laboratory exercises that students were able to complete successfully, as well as by the number of points achieved on the final examination. The statistical analysis of these results, using a T-test, showed that in both regards, the results were statistically significantly better when the proposed solution was employed.

The development of the proposed solution has not yet been completed. On the contrary, the achievement of the presented state constitutes the basis for the further development and incorporation of additional features, both general purpose ones and those specific to the teaching process.

One of the first things to be considered when planning future development is to enable students to access the development environment via the Internet. Such an application would eliminate a significant number of issues that the students face upon setup of the development environment on their private computers. However, although the functional requirements for the use of the environment through the Internet have already been fulfilled, their introduction requires security aspects and risks to be considered. Using the presented solution via an Internet network is particularly important given the growing popularity of on-line learning and massive open online courses.

Enabling debugging using the appropriate protocol and extensions such as the Xdebug have been envisioned as the next significant improvement. The ability to integrate profiling functions and the visualization of obtained data is probably no less important.

Finally, the authors find that this solution constitutes an excellent basis for the integration of the Problem Based Learning (PBL) strategy into the teaching process. This would allow students to receive new, automatically generated project requirements during the project each time. In addition, students would be able to access the system-generated solution, as well as to obtain feedback on the errors they made in their own solution. This approach proved extremely effective [31] in an example of the Java programming language.

Acknowledgment

Authors of the paper are members of projects TR32054, III44006 and ON174008 which are financed by Ministry of Science and Technological Development of the Republic of Serbia.

References

[1] B. B. L. Lim, Teaching Web Development Technologies in CS/IS Curricula, *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)*, Atlanta, GA, 1998, pp. 107-111.

[2] R. Mercuri, N. Herrmann, J. Popyack, Using HTML and JavaScript in Introductory Programming Courses, *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium On Computer Science Education (SIGCSE '98)*, Atlanta, GA, 1998, pp. 176-180.

[3] E. L. Walker, L. Browne, Teaching Web Development with Limited Resources, *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '99)*, New Orleans, LA, 1999, pp. 12-16.

[4] F. Klassner, Can Web Development Courses Avoid Obsolescence?, *Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '00)*, Helsinki, Finland, 2000, pp. 77-80.

[5] S. Murugesan, Y. Deshpande, S. Hansen, A. Ginige, Web Engineering: A New Discipline for Development of Web-based Systems, in: S. Murugesan, Y. Deshpande (Eds.), *Web Engineering, Lecture Notes in Computer Science*, Vol 2016, Springer, Berlin, pp. 3-13.

[6] J. D. Haney, Lessons Learned from Incorporating an XML Component into an Upper Division Web Development Course, *Journal of Computing Sciences in Colleges*, Vol. 18, No. 1, pp. 172-183, October, 2002.

[7] K.-B. Yue, W. Ding, Design and Evolution of an Undergraduate Course on Web Application Development, *ACM SIGCSE Bulletin*, Vol. 36, No. 3, pp. 22-26, September, 2004.

[8] H. J. C. Ellis, An Evaluation of Learning in an Online Project-based Web Application Design and Development Course. *Journal of Computing Sciences in Colleges*, Vol. 21, No. 6, pp. 217-227, June, 2006.

[9] A. M. Gribble, Using Macromedia Breeze in an Advanced Web Development Course Delivered to Face-to-face and Distance Students, *Proceedings of the 5th IASTED International Conference on Web-Based Education (WBE '06)*, Puerto Vallarta, Mexico, 2006, pp. 249-252.

[10] X. Wang, A Practical Way to Teach Web Programming in Computer Science, *Journal of Computing Sciences in Colleges*, Vol. 22, No.1, pp. 211-220, October, 2006.

[11] D. R. Adams, Integration Early: A New Approach to Teaching Web Application Development, *Journal of Computing Sciences in Colleges*, Vol. 23, No. 1, pp. 97-104, October, 2007.

[12] Y. D. Wang, N. Zahadat, Teaching Web Development in the Web 2.0 Era, *Proceedings of the 10th ACM Conference on SIG-Information Technology Education (SIGITE '09)*, Fairfax, VA, 2009, pp. 80-86.

[13] R. W. Kazmerik, Who Cares about Web Services?: A Student Perspective on Web Technology Education, *Proceedings of the 2010 ACM Conference on Information Technology Education (SIGITE '10)*, Midland, MI, 2010, pp. 30-32.

[14] H. S. Al-Khalifa, N. I. Al-Rajebah, Integrating Mobile Web Development into IT Curriculum, *In Proceedings of the 2011 Conference on Information Technology Education (SIGITE '11)*, West Point, NY, 2011, pp. 87-90.

[15] T. H. Park, S. Wiedenbeck. Learning Web Development: Challenges at an Earlier Stage of Computing Education, *Proceedings of the Seventh International Workshop on Computing Education Research (ICER '11)*, Providence, Rhode Island, 2011, pp. 125-132.

[16] R. W. Connolly, Awakening Rip Van Winkle: Modernizing the Computer Science Web Curriculum, *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCSE '11)*, Darmstadt, Germany, 2011, pp. 18-22.

[17] Y. D. Wang, Teaching Web Development at a Distance, *In Proceedings of the 2011 Conference on Information Technology Education (SIGITE '11)*, West Point, NY, 2011, pp. 91-96.

[18] P. Gupta, R. Mata-Toledo, M. Monger, Utilizing ASP.NET MVC in Web Development Courses, *Journal of Computing Sciences in Colleges*, Vol. 27, No. 3, pp. 10-14, January, 2012.

[19] N. A. Hamid, Automated Web-based User Interfaces for Novice Programmers, *Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE '12)*, Tuscaloosa, Alabama, 2012, pp. 42-47.

[20] S. Mudigonda, J. Buerck, The Design of An Undergraduate Web Application Development Course Using Free Technologies and Learning Materials, *Journal of Computing Sciences in Colleges*, Vol. 27, No. 5, pp. 131-139, May, 2012.

[21] C. S. Miller, Metonymic Errors in a Web Development Course, *Proceedings of the 13th Annual Conference on Information Technology Education (SIGITE '12)*, Calgary, Alberta, Canada, 2012, pp. 65-70.

[22] X. Wang, J. C. McKim, The Opportunities and Challenges to Teach Web Programming in Computer Science Curriculum CS2013, *Journal of Computing Sciences in Colleges*, Vol. 29, No. 2, pp. 67-78, December, 2013.

[23] L. Lancor, S. Katha, Analyzing PHP Frameworks for Use in a Project-based Software Engineering Course, *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*, Denver, CO, 2013, pp. 519-524.

[24] R. F. Dugan, Jr., A Single Semester Web Programming Course Model, *Journal of Computing Sciences in Colleges*, Vol. 29, No. 1, pp. 26-34, October, 2013.

[25] X. Liu, Streamlining Courses for the Web Development Concentration of An IT Bachelor Degree Program, *Proceedings of the 15th Annual Conference on Information Technology Education (SIGITE '14)*, Atlanta, GA, 2014, pp. 91-92.

[26] R. Hoar, The Real World Web: How Institutional IT Affects the Delivery of A Capstone Web Development Course, *Proceedings of the Western Canadian Conference on Computing Education (WCCCE '14)*, Richmond, Canada, 2014, Article No. 5.

[27] S. Frees, A Place for Node.js in the Computer Science Curriculum, *Journal of Computing Sciences in Colleges*, Vol. 30, No. 3, pp. 84-91, January, 2015.

[28] G. Šimić, The Multi-courses Tutoring System Design, *Journal of Computer Science and Information Systems*, Vol. 1, No. 1, pp. 141-155, February, 2004.

[29] J. Boyd, Next for DARPA: "Autocomplete" for programmers, http://news.rice.edu/2014/11/05/next-for-darpa-autocomplete-for-programmers-2/.

[30] http://codemirror.net/.

[31] G. Shimic and A. Jevremovic, Problem-based Learning in Formal and Informal Learning Environments, *Interactive Learning Environments*, Vol. 20, No. 4, pp. 351-367, August, 2012.

Biographies

Aleksandar Jevremovic received the B.Sc., M.Sc. and Ph.D. degrees from Singidunum University, Serbia. He is an Associate Professor of Computer Network Security at Singidunum University. So far, he has authored/co-authored number of research papers and made contributions to three books on computer networks, computer networks security and Web development. He is recognized as an Expert Level Instructor at Cisco Networking Academy program.

Nenad Ristic education consists of Bachelor degree acquired 2009 on Sinergija University in field of Programing, Master degree ac-quired 2010 on Singidunum University in field of Advanced Information Technologies. Currently Ph.D. student at Sin-gidunum University in field of Advanced Security Systems. Nenad Ristic's places of employ concluding with current employ-ment are Graduate Teaching Assistant at Sinergija University, April 2009 – Present, Program Administrator, Microsoft IT Acad-emy at Sinergija University January 2011 – Present, and Head IT, Sinergija University, March 2007 – Present. Fields of interest are Computer networks, Web development, Software engineer-ing, Computer Science Education.

Mladen Veinovic received the B.Sc., M.Sc. and Ph.D. degrees in 1986, 1990, and 1996. Respectively, all from the Faculty of Electrical Engineering, University of Belgrade, Serbia. He was born on January 01 1962. From 1987 to 2005, he worked in the Institute of Applied Mathematics and Electronics, Belgrade, where he was head of the Department for speech processing. Since 2005. he has been with Singidunum University, where he is currently rector. He is the author of seven books and a number of journals and conference scientific papers. His current research interests include security, computer networks, data bases and digital signal processing.

Goran Shimic is the Director of the Center for Simulations and Distance Learning, Military Academy, University of Defense, Belgrade, Serbia (http://www.va.mod.gov.rs/cms/view.php?id=21662). Goran is also an Associate Professor of Computer Science specialized in expert systems, object-oriented design and im-plementation, and business software applications. His current research interests are in the area of interoperability between different learning resources & systems, and their integration with diverse types of Web services and applications. So far, he has authored/co-authored about 50 research papers and made contributions to seven books. He is a member of the GOODOL-DAI research network (http://www.goodoldai.org/goran_simic).

Natasa Stanisic holds an MBA Degree from Lincoln University, California. Currently, she is pursuing a Ph.D. at Singidunum University. So far, she has co-authored several research papers and has participated in international conferences. Her research interests are related to computer applications in education, distance learning systems and online language-learning platforms.