Teaching Web Development in the Web 2.0 Era

Ye Diana Wang
Applied Information Technology Department
George Mason University
10900 University Blvd. MS 4F5
Manassas, VA 20110, USA
1-703-993-9288
ywangm@gmu.edu

Nima Zahadat
Applied Information Technology Department
George Mason University
10900 University Blvd. MS 4F5
Manassas, VA 20110, USA
1-703-993-9288
nzahadat@gmu.edu

ABSTRACT
We are experiencing a qualitatively different era of the Web, as evidenced by most Web applications and websites that have emerged and transpired over recent years. The new Web 2.0 era and the technologies beneath it provide both demands and challenges for IT educators to reform traditional Web Development courses. This paper presents our 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. It aims to fill the gap of lacking literature by providing instructional details of our pedagogical strategies and approaches to the design and delivery of Web development courses in such a unique time in computing history.

Categories and Subject Descriptors
K3.2 [Computers and Education]: Computer and Information Science Education – Information Systems Education

General Terms
Design, Experimentation, Languages

Keywords
Web Development, Web 2.0, AJAX, Information Technology Education, Pedagogy

1. INTRODUCTION
The World Wide Web was first started in early 1990s as a work tool for scientists at CERN and then developed into a vast information repository that provided access to a large number of users. That state of the Web was considered as the “read-only” or “brochure Web” because it was used mainly as a medium for one-way information dissemination and allowed very little user interaction [1]. Starting around 2003, we have noticed a shift in how people and businesses were using the Web and how services and software applications were designed and delivered. This trend has been described as the “Web 2.0”—a “second phase” or a new, “improved” Web [2].

Beneath Web 2.0 is a set of Web technologies, which have either recently emerged or moved to a new stage of development with the introduction of RIAs (Rich Internet Applications). One of the most prominent technology and quasi standard for the Web is AJAX (Asynchronous JavaScript and XML), along with some other technologies such as RSS (Really Simple Syndication), widgets (plug-in modular components), and Web services (e.g., SOAP and REST). These technologies are the driving force for the evolution of the current Web.

This rapid advancement of Web technologies has created ever increasing demands from both students and employers for instruction in the techniques necessary to develop and manage systems of the transformed Web. On the other hand, they also present new challenges for IT educators to develop courses related to the teaching of Web development. Many educators struggle to find effective solutions to help students more readily and easily develop the next generation of Web applications. In addition, literature on instruction of Web development in the Web 2.0 era is lacking: the very majority focuses on the exploration of the
We provide some answers to these questions in this paper from an IT educator’s perspective and aim to fill the gap of lacking literature in this area. The rest of the paper is organized as follows. First, the paper describes some challenges and problems of teaching Web development faced by many educators. The next two sections present our approach and strategies in teaching a sequence of two high-level Web development courses and provide details on the courses. The last section discusses the results of the courses and draws some conclusions.

2. CHALLENGES

The explosive growth of Web 2.0 technologies present some significant challenges for today’s IT educators to develop and teach Web development courses. The resulted paradigm shift has a direct bearing on the nature of instruction provided to students who are accustomed to the digital environment. Under these circumstances traditional teaching approaches are showing signs of being inadequate and are unable to prepare students effectively for a transformed Web as well as changed demands from the industries. The approaches and curriculum adopted by most Web development courses suffer some common issues and problems:

Firstly, many educators take an in-breadth approach to cover a wide array of Web technologies and concepts without providing an in-depth introduction to any specific topic. A survey of several courses related to Web design and development [9, 10, 11, 12, 13] indicate that many of them include coverage of several technologies for both client-side development (e.g., HTML, CSS, JavaScript,) and server-side development (e.g., Perl, PHP, ASP.NET, Java), as well as multiple database technologies (e.g., SQL, Java DB, MySQL) and supporting platforms (e.g., Microsoft’s .NET framework, Sun’s Java/J2EE, the open source platform LAMP). Nearly all of the courses do not cover Web development deeply, and some devote much time on other Web-related topics, such as Web server administration, security, privacy and ethics. Although the decision to keep abreast of industry practice is beneficial, many students find it difficult to absorb so much material in a very short amount of time. Consequently, the wide coverage of the technologies and their complexity mitigate the students’ ability to grasp and understand the core body of knowledge being disseminated in the course [14].

Secondly, some topics and technologies covered in Web development courses are outdated and do not address the current Web development trend. As many educators have discovered, one of the biggest difficulties for teaching Web development is to keep up with the quickly changing Web technologies and applications. This is especially critical in the current Web 2.0 era. A course that includes obsolete concepts and unpopular technologies (such as ASP or DHTML) as a major part of the curriculum is unable to train students into becoming developers who are well versed in the latest Web technologies and techniques and who want to keep up with the current Web advancements. Also, the latest generation of undergraduates have already lived in this digital era and been surrounded by the latest Internet technologies. They are a more capable student body and have high educational expectations. As depicted by [15], “Digital Immigrant instructors, who speak an outdated language (that of the pre-digital age), are struggling to teach a population that speaks an entirely new language.”

Lastly but not the least, the majority of educators still maintain an educational pedagogy that is highly teacher-directed, content-centered, and didactic in nature [16]. The popular lecture-by-professor format is inadequate for the instruction of developing Web applications and incompatible with the technology-driven world that we now live in. In such traditional learning environments, students are familiar with being told explicitly what to do and how to do it, rather than engaging in activities to develop their own understanding and acquire solid development skills. Furthermore, some courses use online environment solely for presenting or delivering materials to learners. Such practice arguably adds no real value to student learning, and does not justly the purchase or maintenance of expensive new online education systems [17].

3. AN IMPROVED APPROACH

As revealed in the previous section, it is challenging to teach Web application development, especially in an era that is going through paradigm shift. This is a somewhat unique situation but also presents interesting opportunities. The Web has changed, and it requires us, the IT educators, to adapt quickly and embrace change in our teaching in terms of choice of pedagogy, topics, and many other aspects. In response to those challenges, we have redesigned the curriculum of two Web development courses and delivered them using an improved approach over the last three semesters. Specifically, our approach encompasses some significant elements and improved strategies over the traditional approach of teaching Web development:

(1) Concentrating on AJAX and its enabling technologies and shifting the focus of Web development to user interaction

AJAX is the fundamental technology for Web 2.0, and therefore, it is critical for students to learn to use the specific set of technologies that AJAX encompasses and recognize the enormous power that can be derived from its employment in developing the latest Web applications. Specifically, the client-side of AJAX applications is written in XHTML and CSS and uses JavaScript to add functionality to the user interface. The content of a page is modified using the Document Object Model (DOM), and XML is used to structure the data passed between the server and the client. The AJAX component that manages interaction with the server is provided by the XMLHttpRequest object (XHR). The server processing can be implemented using any
server-side technology, such as ASP.NET. Syndication and remixing of content is usually accomplished through the use of RSS. Web services, which make it possible to compose services from many different websites, are a critical part of the infrastructure used by Web 2.0 applications based on the implementation of the SOAP or REST model.

In the courses we have explicitly selected these AJAX enabling technologies and cover them in depth. We have also emphasized the importance of building Web applications that are more focused on user interaction and facilitate a “richer” user experience than traditional Web applications. This focus of Web development is consistent with the nature of Web 2.0. Students not only learn about leading edge technologies to create more interactive and dynamic websites but also adapt into the new Web development model that is centered on user interaction.

(2) Breaking into a two-course sequence, covering both client-side and server-side Web development

To avoid overwhelming students with too many topics in one class, we have divided the instruction of Web development into a sequence of two high-level courses, namely Web Development I (WDI) and Web Development II (W DII). The courses, each of which was originally offered to cover a disconnected set of Web technologies, have been redesigned completely. Web 2.0 applications are client-server applications and this has helped determine the structure of the two courses. WDI covers the component technologies that enable client-side AJAX programming and interface features; W DII covers server-side technologies for the implementation of more advanced AJAX-based Web applications, as well as technologies for the integration and communication between the client and the server. In this way, students are provided with a smooth and complete instruction on developing current Web applications from the front-end to the back-end.

(3) Assigning projects that integrate covered topics and deliver real-world applications

Web 2.0 is about finding new ways to make a number of existing and evolving technologies work together [18]. Because of the importance of integration, we have designed course projects that require students to use a combination of covered topics and technologies to develop Web applications. WDI adopts an “integrate early, iterate often” strategy as proposed in [19] to assign four projects throughout the course. Students are expected to implement a functional website with navigation features using XHTML and CSS in the first project. Each subsequent project builds on the previous project and requires the use of newly covered component technologies of AJAX to add functionalities or features. By the end of the semester students complete an attractive and dynamic website with AJAX features. In W DII students are required to build two separate working projects, one with focus on the design and the other with focus on the implementation of a database-driven, AJAX-based Web application that incorporates several advanced functionalities and server-side technologies.

The projects for both courses place emphasis on delivering real-world Web applications. Students are encouraged to be creative in their choice of topics and design in developing Web 2.0 applications and websites. Examples of student work include a Web-based tutoring system, a vendor website with e-commerce capabilities, an online photography portfolio, and a community college portal. Realistic contexts give the students a concrete motivation to learn the necessary concepts and skills required to complete a course project. In addition, students can understand the intricacies of developing sophisticated real-world Web applications and appreciate the value of the materials covered.

(4) Using constructivist teaching methods

Knowing the drawbacks and inadequacy of the traditional teacher-directed lecturing approach for today’s IT education, we have used some constructivist teaching methods in the delivery of instruction. Constructivism is based on the premise that knowledge cannot be transmitted but has to be constructed by the individual [20]. Therefore, we have transformed the classroom into a student-centered environment by keeping students engaged in hands-on exercises and solving relevant problems during class. Both courses are conducted in a laboratory setting where each student has immediate access to a computer with the required programs.

In WDI each class usually starts with a short introduction by the instructor on the particular syntax and semantics that are native to the technologies covered. During the rest of the class students examine some live-code examples with the instructor and are challenged to complete small program segments frequently. Similarly, in order to maximize learning effectiveness, no PowerPoint slides is used in W DII. Instead, actual exercises and examples are done on the overhead with the students following along. Following the exercises, students are given their own practice quizzes and hands-on material to complete. In both courses teaching is conversational and interactive in which instructors prompt students to ask questions and often give personal attention to individual student struggling with some code with the help of a teaching assistant. By facilitating such active learning, in which the students seek solutions to the given problems, the instructors serve as an initiator and an adviser in the learning process, rather than founts of knowledge.

In summary, these elements make for a suitable and effective approach to teaching Web development in the Web 2.0 era. The next section discusses the details of these two courses, the experience gained, and our efforts in addressing some issues.
4. COURSES’ DETAILS

WDI and WDII are each a three-credit-hour high-level course in our BSIT (Bachelor of Science in Information Technology) program and a standard lecture/lab course of 15 weeks. They are offered as part of the Multimedia & Web Development Concentration courses, but students from other concentrations (e.g., Database Technology & Program, Information Security) can take them as an elective concentration course. Both courses, each of which originally covered a disconnected set of Web technologies and topics, have been redesigned and delivered using our improved approach over the past three semesters.

Students attending WDI and WDII tend to be young (early twenties) with a huge appetite for learning Web development skills. They have typically been attending university for at least three years and are getting ready to graduate. They are eager to join the work force and obtain well paying jobs.

4.1 Prerequisites

There are three low-level IT courses as prerequisites for WDI. Students taking the course should have previous experience in creating and publishing a website that includes multimedia components and using a Web authoring tool (e.g., Dreamweaver) and graphic editing tool (e.g., Photoshop), understand fundamental Web design principles, and have basic problem solving skills with procedural programming or scripting techniques.

The prerequisite for WDII is WDI. Students usually take the courses in two consecutive semesters. In addition to the knowledge and experience gained from WDI for client-side Web development, students taking WDII also should have a basic understanding of the Internet client-server architecture and relatively strong Web programming and scripting skills.

4.2 Topics

4.2.1 Web Development I (WDI)

The focus of WDI is on client-side Web development. Specifically, the course begins with an in-depth introduction on the specifications and use of XHTML and CSS. Students learn how to create and style different page elements (e.g., list, table, form, navigation bar, etc.) and build both hybrid and CSS-base page layouts. The course then moves into JavaScript and gives a systematic walkthrough of the language (e.g., control structures, functions, arrays, objectives, events, etc.) and DOM. Students learn how to add interactivity to Web page, to personalize Web interface by using cookies, implement interesting features (e.g., a random image generator, online quiz, AJAX form validation), and dynamically modify page elements using DOM and its collections. The final weeks of the course move to the topics of XML, RSS, XSL (Extensible Stylesheet Language) and the integration of all covered technologies. Students learn how to transform XML files into Web pages, add syndication by using RSS, and more importantly, incorporate several technologies to implement AJAX functionalities (e.g., dynamic data regions, partial page update) on Web pages.

In addition, the course provides instruction on the use of the latest addition to Adobe Dreamweaver—the Spry Framework, which is implemented as a set of JavaScript and CSS libraries and tools for AJAX features. Students learn to add smooth and graceful transitions to page elements and implement dynamic and custom components (e.g., accordion, collapsible panel, etc.) by using Spry effects and widgets.

4.2.2 Web Development II (WDII)

The focus of WDII is on server-side Web development, it builds on the material from WDI. The course begins with Web servers (Apache and IIS) and how they are used. This discussion is centered in setting up Apache and IIS, introductory Web server concepts and security, virtual directories, and interaction of server-side languages with these Web servers. It then moves into SQL and use of SQL Server in particular; MySQL is also taught and discussed. Next the course moves into Web application development using ASP.NET and AJAX. These topics take up the bulk of the class. During this phase, ASP.NET forms, controls, objects, validation, events, code-behind, programming (Visual Basic or C#), and security are taught and discussed. Advanced CSS design techniques are taught as required for WDII. While LINQ is discussed, the bulk of the database related queries are still done using SQL in order to be more compatible with the rest of the industry. Web services are touched upon with specific examples.

It is worth mentioning that the course used to cover Java related topics such as JSP and Servlets prior to the redesign of the course. These topics have been omitted because the department is now offering two undergraduate courses in Java. Given the limited time in the course, we find it necessary to give an in-depth treatment on the popular .NET framework and related Microsoft technologies to satisfy student’s demand and interests on these topics.

4.3 Textbooks and Supplemental Material

The current textbook used is *Internet & World Wide Web—How to Program* by Deitel [18]. WDI uses the first half of the book and WDII uses the second half. The book is very comprehensive and covers a great many technologies for developing modern Web 2.0 websites. It does, however, discuss certain particular technologies in great detail, such as XHTML, CSS, JavaScript, AJAX, and ASP.NET. This book is widely used in courses that teach pure client-side Web programming, courses that teach pure server-side Web programming, and courses that mix and match some client-side and some server-side Web programming.

In addition to the textbook, we also use many of the excellent resources found on the Internet. For WDI, a resource page is made accessible on the Blackboard companion course site and has been compiled and constantly updated to provide references (e.g., language specifications at the W3C website), tools (e.g., RSS validating services), and techniques (e.g., CSS hacks) for Web development. We also present our own lecture slides that have been developed based on many various resources. For WDII, video tutorials are used for the ASP.NET programming section. This allows the students to view the videos at any time. The tutorials are provided by [http://www.asp.net/](http://www.asp.net/) and are
areas for each project are summarized in the following table:

### 4.4 Computer Lab

Both courses are conducted entirely in a laboratory setting. The lab is equipped with 40 computers with large flat-screen monitors and high-speed Internet. Each computer is installed with the software and platform required for the Web application courses, such as the current version of Web browsers, IIS, Adobe Dreamweaver, and Visual Studio. Students are allowed to install any tools they wish. The computers are automatically reset to a standard configuration each day and all the student installations and files are removed. Students can also use their own external hard disks in conjunction with the Virtual PC program for creating a virtual programming environment and saving their files and projects. This gives the instructor and the students the flexibility needed to experiment without worrying about the computer configuration.

### 4.5 Assessment

#### 4.5.1 Projects

Both courses place heavy weights on projects (40% of the final grade). WDI assigns four projects that are targeted at the common goal of building a functional and interactive website that includes attractive interfaces and AJAX features and span the entire semester. The projects integrate all covered technologies into the final website, with one project building on top of the previous one. Through the development of a “real-world” website on a self-selected, realistic topic and in an continually updated fashion, students gain valuable practice in seeking solutions to challenging problems and adapting to the current way of Web application development and delivery. The specific requirements and focus areas for each project are summarized in the following table:

<table>
<thead>
<tr>
<th>Project</th>
<th>Focus Areas</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>XHTML &amp; CSS</td>
<td>CSS-styled page elements (lists, links, images, tables, forms, navigation bars), external style sheet, accessibility features, etc.</td>
</tr>
<tr>
<td>Project 2</td>
<td>JavaScript</td>
<td>Functions, arrays, cookies, random image generators, Date object, interactive greeting, personalized interface, etc.</td>
</tr>
<tr>
<td>Project 3</td>
<td>DOM &amp; Events</td>
<td>Dynamic modification of page elements and styles, events, rollover effects, AJAX form validation, etc.</td>
</tr>
<tr>
<td>Project 4</td>
<td>XML &amp; RSS</td>
<td>RSS feed, XSL style sheet, customized RSS reader, widgets, dynamic data fields, etc.</td>
</tr>
</tbody>
</table>

Coding requirements with regards to comments and readability of the programs are strictly enforced. Also, each project requires detailed documentation written in the form of a “user guide” or “product manual”, which accounts for a portion of the grade.

Students must publish their project websites on the Web server supported by the university. In addition, a project presentation is required at the end of the semester. Students may benefit the opportunity to showcase their work, receive constructive feedback from the instructor and classmates, and practice independent presentation skills.

In WDI, students are required to build two working projects, each a complete website, database-driven, using advance CSS (for screen, print, and handheld devices), with ASP.NET and AJAX technologies. These websites allow for client login, lost password recovery, access to privileged areas of the site, fetching of data from databases, and presentation of customized pages using server-side programming. Both projects focus on getting Web users to interact with the site, encouraging Web 2.0 concepts. Server-side AJAX is introduced between Project 1 and Project 2, allowing the students to see how partial page updates can enhance user experience on a Web page. A short description of each project is provided below:

- **Project 1**: is creating a Pets website using ASP.NET. The focus is on design, using ASP.NET, CSS, XHTML, and JavaScript to create an attractive website. Creativity is highly encouraged. Students are not required at this point to incorporate server-side technology or database-driven pages. Code-behind is used at this stage but the primary focus is on design.
- **Project 2**: serves as the final project, creating a Web application or portal for a Tahoe Community College in California. This project incorporates all the requirements of the Pets project and it also requires server-side technology and ASP.NET AJAX to create database-driven pages with secure login, access to privileged pages, use of server-side cookies, server-side validation, and presentation of dynamic page content.

#### 4.5.2 Assignments and Exams

The assignments for both courses are similar in terms of format and include required reading and watching of videos, hands-on coding, and in-class exercises. These assignments are mostly doable in a few days or within the class: some are aimed at getting students prepared and familiarized with the topic in focus; some are aimed at reinforcing materials taught in class and stressing problem solving by having the students write many short problems and include required reading and watching of videos, hands-on coding, and in-class exercises. These assignments are mostly doable in a few days or within the class: some are aimed at getting students prepared and familiarized with the topic in focus; some are aimed at reinforcing materials taught in class and stressing problem solving by having the students write many short problems; and some are aimed at getting immediate feedback from students in order to provide further clarification and elaboration of the materials.

Examples of lab assignments given in WDI are:

- Implementing a list-based navigation bar using CSS and XHTML
- Converting a hybrid-layout Web page to a CSS-layout Web page
- Writing a JavaScript program that can split telephone numbers inputted by users
- Creating a Web page where users can dynamically change the styles of page elements (e.g., background, text) by typing the style name into a dialog box

Students must publish their project websites on the Web server supported by the university. In addition, a project presentation is required at the end of the semester. Students may benefit the opportunity to showcase their work, receive constructive feedback from the instructor and classmates, and practice independent presentation skills.

In WDI, students are required to build two working projects, each a complete website, database-driven, using advance CSS (for screen, print, and handheld devices), with ASP.NET and AJAX technologies. These websites allow for client login, lost password recovery, access to privileged areas of the site, fetching of data from databases, and presentation of customized pages using server-side programming. Both projects focus on getting Web users to interact with the site, encouraging Web 2.0 concepts. Server-side AJAX is introduced between Project 1 and Project 2, allowing the students to see how partial page updates can enhance user experience on a Web page. A short description of each project is provided below:

- **Project 1**: is creating a Pets website using ASP.NET. The focus is on design, using ASP.NET, CSS, XHTML, and JavaScript to create an attractive website. Creativity is highly encouraged. Students are not required at this point to incorporate server-side technology or database-driven pages. Code-behind is used at this stage but the primary focus is on design.
- **Project 2**: serves as the final project, creating a Web application or portal for a Tahoe Community College in California. This project incorporates all the requirements of the Pets project and it also requires server-side technology and ASP.NET AJAX to create database-driven pages with secure login, access to privileged pages, use of server-side cookies, server-side validation, and presentation of dynamic page content.

### 4.5.2 Assignments and Exams

The assignments for both courses are similar in terms of format and include required reading and watching of videos, hands-on coding, and in-class exercises. These assignments are mostly doable in a few days or within the class: some are aimed at getting students prepared and familiarized with the topic in focus; some are aimed at reinforcing materials taught in class and stressing problem solving by having the students write many short problems; and some are aimed at getting immediate feedback from students in order to provide further clarification and elaboration of the materials.

Examples of lab assignments given in WDI are:

- Implementing a list-based navigation bar using CSS and XHTML
- Converting a hybrid-layout Web page to a CSS-layout Web page
- Writing a JavaScript program that can split telephone numbers inputted by users
- Creating a Web page where users can dynamically change the styles of page elements (e.g., background, text) by typing the style name into a dialog box

Students must publish their project websites on the Web server supported by the university. In addition, a project presentation is required at the end of the semester. Students may benefit the opportunity to showcase their work, receive constructive feedback from the instructor and classmates, and practice independent presentation skills.

In WDI, students are required to build two working projects, each a complete website, database-driven, using advance CSS (for screen, print, and handheld devices), with ASP.NET and AJAX technologies. These websites allow for client login, lost password recovery, access to privileged areas of the site, fetching of data from databases, and presentation of customized pages using server-side programming. Both projects focus on getting Web users to interact with the site, encouraging Web 2.0 concepts. Server-side AJAX is introduced between Project 1 and Project 2, allowing the students to see how partial page updates can enhance user experience on a Web page. A short description of each project is provided below:

- **Project 1**: is creating a Pets website using ASP.NET. The focus is on design, using ASP.NET, CSS, XHTML, and JavaScript to create an attractive website. Creativity is highly encouraged. Students are not required at this point to incorporate server-side technology or database-driven pages. Code-behind is used at this stage but the primary focus is on design.
- **Project 2**: serves as the final project, creating a Web application or portal for a Tahoe Community College in California. This project incorporates all the requirements of the Pets project and it also requires server-side technology and ASP.NET AJAX to create database-driven pages with secure login, access to privileged pages, use of server-side cookies, server-side validation, and presentation of dynamic page content.
Creating an XML document based on the information for a DVD library and displaying it as an XHTML table with the use of an XSLT style sheet

Examples of in-class exercises given in WDII are:
- Installation of Apache and IIS including configuration, virtual directories, http access, security, etc.
- Installation of SQL Server and MySQL, writing SQL queries, CRUD concept
- Learning to use Visual Studio 2008 IDE
- Learning ASP.NET server concepts
- Incorporating AJAX into ASP.NET

The midterm and final exams of WDI and WDII are rather traditional and mostly multiple-choice questions. The exams are designed to assess a student’s theoretical understanding of the covered materials in class and problem solving skills in the particular Web programming or scripting languages. Some questions require students to trace through the code and figure out what the output will be.

### 4.6 Addressing Issues

This section summarizes issues that come up and how they are addressed in our courses. As is the case in just about every IT field, security and accessibility are the primary (and opposing) issues that must be dealt with. We address them below.

#### 4.6.1 Security and Privacy Issues

Part of the Web 2.0 challenge is security and privacy. Students in WDI and WDII are taught the importance of security and respect for privacy of others. Both classes explore security and privacy issues and how to handle them. Secure http (https) is covered and secure access to Web sites using cookies and sessions are taught and made requirements during the project phases of both courses. In WDII, encryption of data stored in databases is also covered. In particular, the use of secure hash algorithms such as MD5 and SHA-1 are discussed and integrated into the course.

#### 4.6.2 Accessibility Issues

Web 2.0 encourages user participation and part of that participation comes from mobile devices, users with various handicaps, and international users. Through proper use of CSS, XHTML, and AJAX, the two courses instruct students on creating attractive pages that are accessible nearly universally to everyone who may visit. This is of particular importance in the US as the government requires its websites to be compliant for handicap access.

#### 4.6.3 Rapidly Changing Web Technologies

It can be difficult for the instructor and students to have to constantly upgrade or change their skills but also rewarding. We attempt to provide the latest technologies for our student body in WDI and WDII. This does put pressure on us as instructors to have to learn the latest ourselves, and learn well enough to teach that material in a cohesive manner to our students. With such busy schedules, it is at times challenging to learn the newest and latest. The approach that seems to work well is learning one technology or piece of that technology at a time, building over time. This makes learning easier, more fun, and an almost daily ritual to learn one little new technology. In turn we continuously refine the curriculum of the courses with newly acquired technologies and improved techniques. Over the past three semesters during which the courses have been redesigned and delivered, we have taken an incremental approach to incorporate new materials into the courses.

### 5. CONCLUSIONS

The goal of WDI and WDII is to help students gain confidence in developing advance Web applications with the latest technologies and be marketable upon graduation. The courses have been successful both in terms of the accomplishment of the course goal and in terms of the quality of work produced. Students have built several websites during the duration of the courses, some in class, and some as their projects. The websites developed are indeed advance and utilizing important leading edge technologies. These websites also serve as their personal portfolios, which are used to impress prospective employers, or references for future development work. Students completing the courses with a B or better are well prepared for careers in advanced Web development.

In conclusion, Web courses need change just as frequently as the Web itself is changing. We have focused on Web 2.0, but as new technologies and paradigms are introduced, regardless of the nomenclature used to describe them, courses focused on Web will need to be updated and the educators will need to adapt. We believe it is important not to put too much in a single class, and at times, it may be necessary to simply add an additional class to allow all needed topics to be covered in depth. It has been our experience that students learn Web technologies best by “doing”. Our job as educators is not only to teach the proper technologies and explain the paradigms but also to foster creativity in design and development. While teaching Web development is always a challenge due to the ever changing world of the Web, it is certainly never boring. That positive outlook of the very changing, improved, enhanced world of the Web is what we strive to bring to our classes and our students.

### 6. REFERENCES


