

Enabling Customization through Web Development: An Iterative Study of the Dell Computer Corporation Website

Chang Liu

Brian G. Mackie

Operations Management and Information Systems

College of Business

Northern Illinois University

Dekalb, IL 60115, USA

cliu@niu.edu bmackie@niu.edu

ABSTRACT

Throughout the last decade, companies have increased their investment in electronic commerce (EC) by developing and implementing Web-based applications on the Internet. This paper describes a class project to develop a customized computer website which is similar to Dell Computer Corporation's (Dell) website. The objective of this project is to ensure that students receive real-world experiences in the classroom. Students went through three rounds of iterative study and improvements to finalize the project design. This paper presents this customization process using the Microsoft Visual Studio 2005 platform. The paper concludes with a discussion of "lessons learned" and suggestions for effectively teaching project development in an EC design course.

Keywords: Design, Customization, Project, ASP.Net, Web development, Iterative Development.

1. INTRODUCTION

The need for integration of education, practice, and information technology is growing (Lu, Yu, and Liu, 2003). Instructors in Management information systems (MIS) and other disciplines are often encouraged to find ways to assist their students in improving their project development skills in an iterative setting as opposed to just having case studies. With the advancement of the Internet and Web technologies, instructors are not only able to demonstrate real business applications online, but also facilitate and guide students through the process of gathering information, analyzing business cases, testing validity and applicability, and creating meaningful solutions for business organizations. The use of the Internet and Web technologies is quickly becoming an educational given, and an important, yet increasingly viable part of a student's learning environment (Landry, Griffeth, and Hartman, 2006). Therefore, it is very appropriate to have students create a customized website by examining and implementing a real online business application in an electronic commerce (EC) design course. This paper demonstrates a student project to develop a computer website which is similar to Dell Computer Corporation's (Dell) website by allowing customers to build their own computers.

2. THE PROJECT

The project was based in a semester-long graduate level EC course taught in the fall of 2006 at a large, Midwestern

university. All students enrolled in this course were MIS majors. The lead author of this paper was the course instructor. The project required the students to develop a Web application that would allow potential customers to select a desktop computer and customize it using similar functionality as available on Dell's website. An iterative approach with three rounds of the project's development was adopted. These three rounds development activities were individual project works with the first round assigned to the students after the completion of the first three weeks of the class.

2.1 First Round: Web Application Replication

In this round of the project, the students were asked to evaluate a particular EC website, Dell, Inc. (<http://www.dell.com>), and evaluate a specific computer system, the Dell Dimension E510. Once they evaluated the Dell computer model and had a basic understanding of it, they were asked to design a Web application similar to Dell's that would handle the selection and customization of the Dimension E510 model. The students were instructed that the website they developed in this round of the project was a "first-of-series" prototype to be used as a pilot with more advanced features added in later rounds. It was explained that the prototyping methodology makes it possible to build a functional website more quickly and then add more functionality after the initial prototype had been developed. To do this, the students designed a Web Form in the Microsoft Visual Studio 2005 to display the basic desktop components including processor, memory, hard

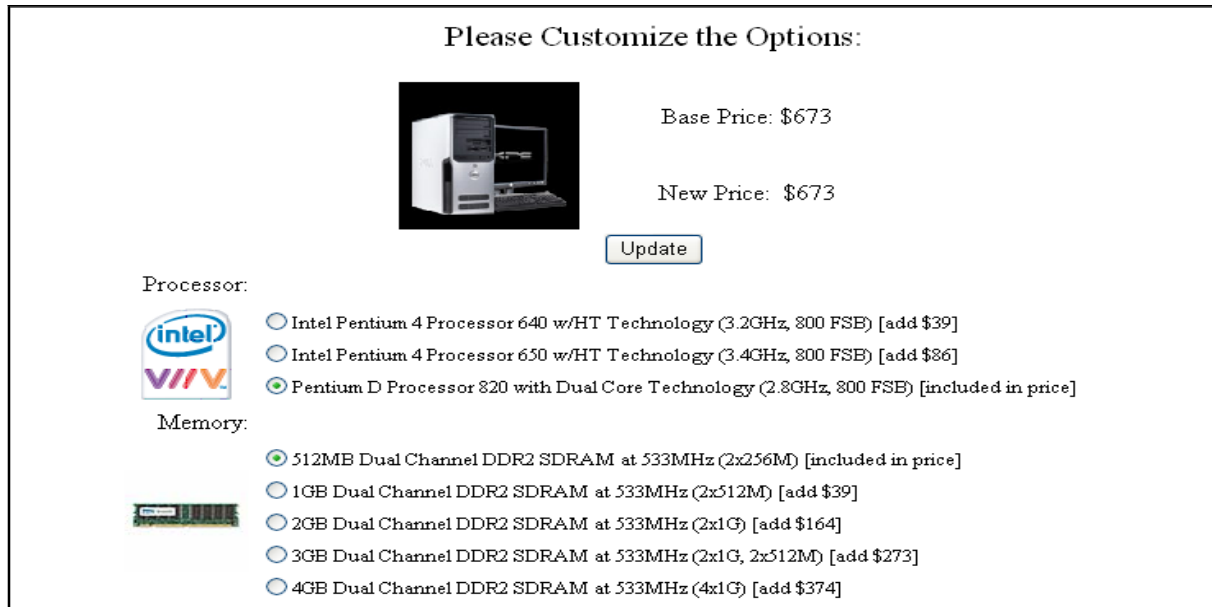


Figure 1: Web page for Customizing Computer Components

drive, CD/DVD drive, operating system, monitor, etc. Figure 1 displays the Web page produced by the students for this round.

Each computer component offered multiple customization options. Those options were implemented using Web Form controls such as radio button and checkbox lists. The students customized the component options by including an “Update” button to allow re-calculation of the computer price once different options were selected. The page was designed using ASP.Net 2.0 technology. ASP.Net 2.0 is the latest server-based technology from the Microsoft Visual Studio 2005 development platform, designed to create interactive and dynamic HTML pages for a website with minimal programming.

The students were instructed to use the Microsoft Visual Basic programming language for the ASP page design. One of the biggest advantages of using ASP.Net 2.0 for Web page design is its simplicity. Microsoft claims that the tools and objects provided in Visual Studio 2005 help ASP.Net 2.0 reach “70% less code” compared with the previous version of the ASP technology (Hart, Kauffman, Sussman, and Ullman, 2006). This allowed the students much more opportunities to explore EC application design activities and dramatically reduced code writing time and efforts (Liu, 2006).

For the Web design, the students simply dragged and dropped several radio button lists and/or a checkbox list onto the Web page, and then typed item texts and values for each customization option. In addition, they were asked to use basic layout design techniques to professionally align images and the other objects used on the page. The “Update” button click action was used to calculate the new price information. After explaining the project assignment, the instructor demonstrated in the class on how to handle radio button and checkbox lists in a Web Form design. Appendix 1 presents the student solution coded for the “Update” button. All

students were required to complete this round of Web application replication in a week. It turned out that the students spent more time on designing the page layout than writing code. At the end of this round of the project, the students delivered a functional website that would allow customization of a specific computer system.

2.2 Second Round: Application Improvement

The students developed the basic prototype in the first round. They were now asked to thoroughly examine the Dimension E510 model offered at Dell’s website. Each student wrote an individual report about the design and page layout of the Dell and gave recommendations on how to improve the Web page developed in round one. The students had a week to finish their reports immediately after the completion of the first round project work. They were asked to concentrate on how to improve their websites in terms of technical quality from a Web performance perspective. In particular, each student was required to write down specific items that need to be changed or added to the Web page to improve the initial prototype developed in round one. No specific guidance was given except for requiring the report in bullet items to specify each improvement recommendation in detail. The authors feel strongly that this approach allowed the students to actively involve with their project design and to promote their critical thinking abilities. Moreover, we believe that the students need to understand that a solid EC application should follow the Three-Tier Architecture design principle, i.e., separating the presentation, business, and data logics to help better maintain the site design. Based on these considerations, the instructor summarized recommendations made in the students’ reports and then proposed the following to improve the project:

- Most students recommended that instead of manually typing in the customization options into the radio button and checkbox lists during the page design, those options

could be stored in a database and all computer options could be automatically pulled from the database. This would allow more flexibility when adding option items or changing option prices. Therefore, the students needed to implement a database and use the basic mechanisms built into ASP 2.0 to access the data from the database and to populate radio button and checkbox lists on the Web page.

- Following the discussion of using a database to store customization options for the Dell, the students were instructed that the Web page design should follow the Three-Tier Architecture design principle. The students learned that they should follow this architecture which is to focus on graphical user interface (GUI) design in the presentation tier, to concentrate on application logic in the business tier, and to handle data storage and data access in the data tier. The students were asked to implement this design principle in this round.
- The database should include realistic price information for each customization option for the Dell Dimension E510 model. However, several students reported that they were not sure about the pricing for each computer component option. They asked how they could get more accurate pricing for many of the customization options. The instructor suggested that students might want to use a website that tracked computer hardware pricing. In this round of the project, the Price Watch website (<http://www.pricewatch.com>) was used to search cost information for each option of E510's computer specification. The students could then include the pricing for each computer component option in the database. This would ensure that the computer price reflected the Dell computer component in the real world.

- Most students mentioned that when they had used Dell's website the price information updated instantly once they selected a given option. This include having the wording of each option, whether it was a radio button or a checkbox, update based on the new selection. For example, if Windows XP Professional was selected, the wording for that option would change from 'Windows XP Professional [add \$120]' to 'Windows XP Professional [included in price]'. All other options for that component would be adjusted to reflect correct wording. The students were instructed how they could implement this feature to update wording display and the price information when a customization option was selected or deselected.

Unlike the first round of the project, the students recommended that the desktop components for processor, memory, hard drive, CD/DVD drive, operating system, monitor, etc. would not be displayed in a list view from top to bottom. All components would be displayed in a separate view (icon view) for better Web layout design.

In this round, the instructor demonstrated in the class on how to handle the "processor" component by following the Three-Tier Architecture design principle. The students were required to complete all other E510's components design after understanding the class demonstration.

The students created an item_e510 data table consisting of the fields ItemID, ItemType, ItemDescription, Cost, and DefaultSelection in a remote Microsoft SQL Server database. Each student used his/her own database table designed in this round to complete the assignment. Figure 2 presents the instructor's own table records showed in the class. The students then concentrated on developing a

ItemID	ItemType	ItemDescription	Cost	DefaultSelection
1	Processor	Intel Pentium 4 Processor 640 w/HT Technology (3.2 GHz, 800F5B)	199	0
2	Processor	Intel Pentium 4 Processor 650 w/HT Technology (3.4 GHz, 800F5B)	235	0
3	Processor	Pentium D Processor 820 with Dual Core Technology (2.8GHz, 800F5B)	154	1
4	Memory	512MB Dual Channel DDR2 SDRAM at 533MHz (2x256M)	36	1
5	Memory	1GB Dual Channel DDR2 SDRAM at 533MHz (2x512M)	66	0
6	Memory	2GB Dual Channel DDR2 SDRAM at 533MHz (2x1GB)	132	0
7	Memory	3GB Dual Channel DDR2 SDRAM at 533MHz (2x1GB, 2x512M))	198	0
8	Memory	4GB Dual Channel DDR2 SDRAM at 533MHz (4x1GB)	264	0
11	HardDrive	80GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache	36	0
12	HardDrive	160GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache	63	0
13	HardDrive	250GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache	114	0
15	HardDrive	DataSafe 160GB (includes main hard drive plus a hidden reserve hard drive)	155	1
16	HardDrive	DataSafe 250GB (includes main hard drive plus a hidden reserve hard drive)	256	0
23	DVD	Single Drive: 48x CD-RW/DVD-ROM Combo Drive	30	1
24	DVD	Single Drive: 16x CD/DVD burner (DVD +/-RW) w/double layer write capability	69	0
28	DVD	Dual Drives: 16x DVD-ROM Drive +16x DVD+/- RW w/double layer write capability	71	0
30	Screen	No Monitor	0	0
31	Screen	15 inch E156FP Analog Flat Panel	30	0
32	Screen	FREE UPGRADE!! 17 inch E176FP Analog Flat Panel	31	1
33	Screen	19 inch E196FP Analog Flat Panel	237	0
34	Screen	17 inch Ultrasharp 1707FP Digital Flat Panel	199	0
35	Screen	19 inch Ultrasharp 1907FP Digital Flat Panel	303	0
36	Screen	20 inch Ultrasharp 2007FP Digital Flat Panel	404	0
37	Screen	24 inch Ultrasharp 2405FPW WideScreen Digital Flat Panel	799	0
41	OS	Genuine Window XP Media Center Edition 2005	199	1
42	OS	Genuine Window XP Media Center Edition 2005 with re-installation CD	209	0
43	OS	Genuine Window XP Home Edition	219	0

Figure 2: Data Table Records for Dimension E510

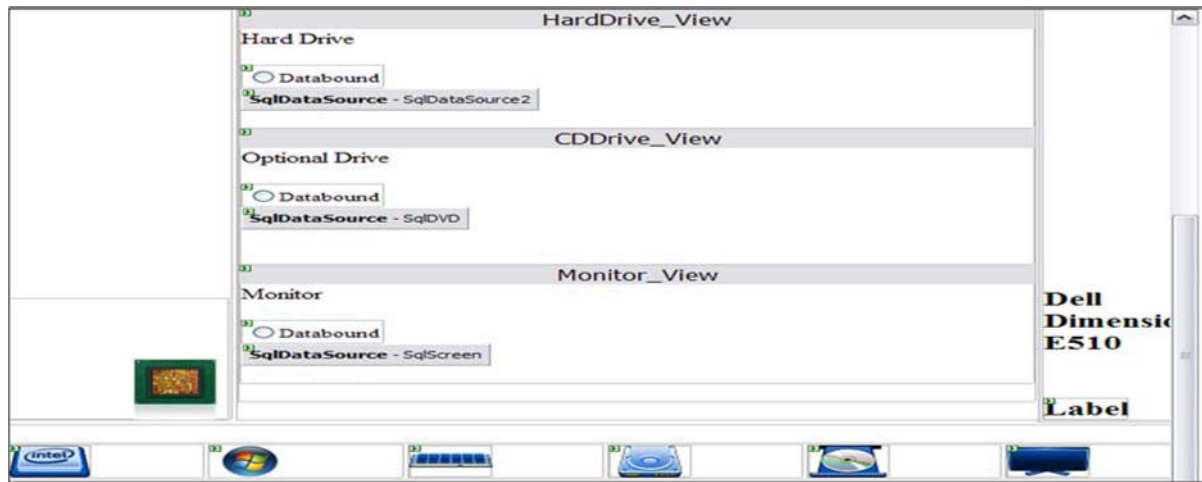


Figure 3: The Page Layout Sample

professional UI for the page layout as shown in Figure 3. The students were asked to complete the layout design with Multiview and View objects in ASP.Net 2.0 to ensure that each desktop component displayed in a separate icon view. Additionally it was recommended that they carefully name the radio button and/or checkbox list controls for each desktop component so that the control names would match the data field names in the backend item_e510 table. It was also recommended that they use a separate SqlDataSource control for each desktop component designed in the Web page. Figure 4 shows how the students configured the SqlDataSource to display appropriate data content in a radio button list and/or checkbox list for each desktop component. Figure 5 presents a browser display using this layout design.

Moreover, they set AutoPostBack properties of the control objects used in the page to True so that the price information and the display wordings for each component option could be updated instantly in the browser. Appendix 2 presents the code finished by the students for enabling each

component option to display correct wording for the content and cost information. This round lasted three weeks. At the end of round two, the students stated that they felt they had made a drastic improvement of their websites. For example, one student stated that “even though I did not like having to do an exercise every week, towards the end of this project assignment I realized that’s the only way it was possible for me grouping all the technical features.”

2.3 Third Round: Final Application Design

After the completion of the second round, the students were excited that they had actually produced a data-driven Web application from a real world example. Each student was again asked to revisit Dell’s website and write a report about how the website could be further improved. In particular, the students were asked to suggest ways to improve the Web design from both technical and user perspectives. The report format was again required in bullet

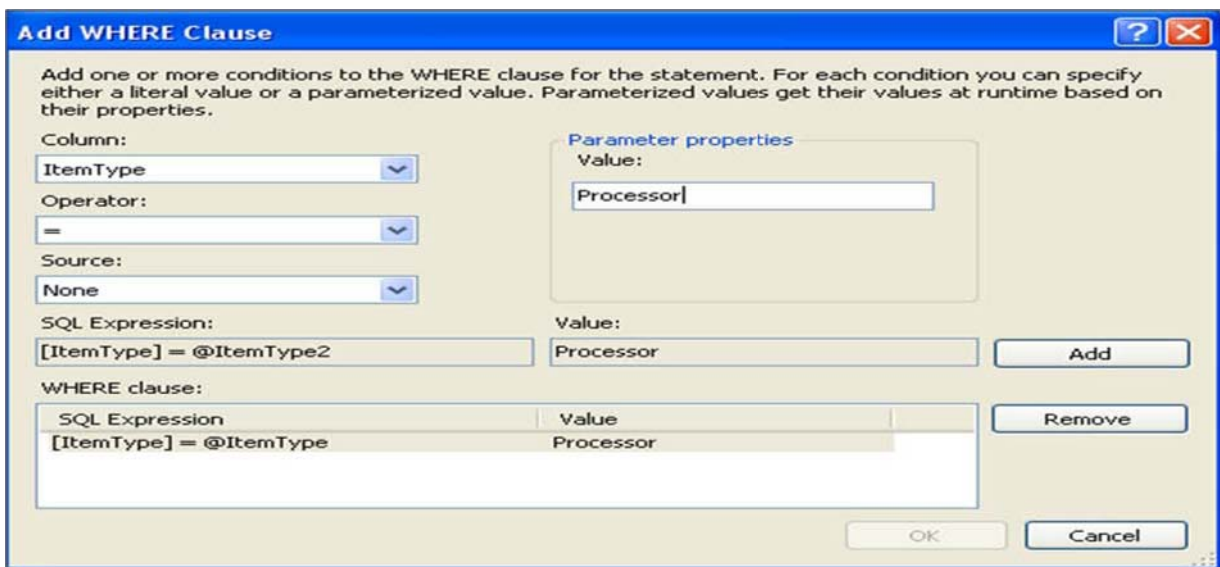


Figure 4: An Example of the Where Clause for a SqlDataSource

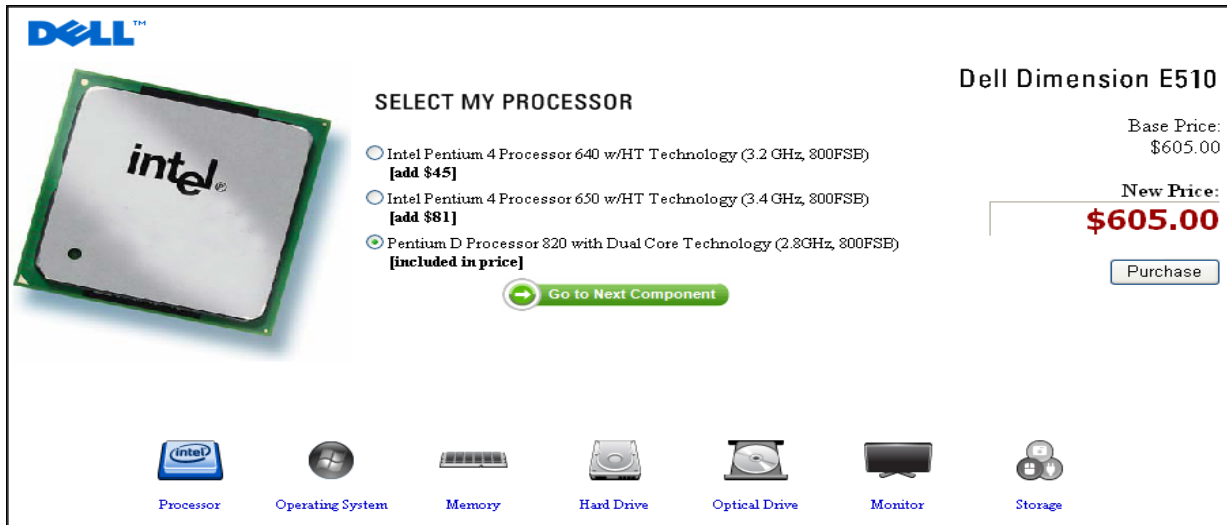


Figure 5: The Web Page Sample for Dell Dimension E510

items to specify each improvement recommendation in detail. From the instructor perspective, the intention of this round of the project was to extend round two activities to ensure the design work in more depth and the students could deliver a professional look and feel EC website. The students had four weeks to complete their final application design and submitted their works at the end of the semester. Here is a summary of the recommended improvements adopted by the students in their final projects:

- Most students suggested expanding the current design to provide more desktop and/or laptop models in the final round. This provided more depth in the technical design and offered more choices to the customers. Therefore, the database was redesign to allow multiple computer models. Figure 6 shows the database diagram. By using three tables, i.e., model, model_item, and item, the students were able to
- expand the design to incorporate different models in the Web application. The students actually implemented two computer models - Dell's XPS410 and Dimension E510 - in the final project.
- Help links were added for each computer component. Customers could follow the links to learn more about the given component such as processor, memory, hard drive, optical drive, monitor, or operating system. It was felt that this feature could help customers make a more informed selection of a specific component option.
- To help customers compare the featured computer model with other customized options, several students recommended to redesign each of the component pages so that the a customer could now view the base configuration of the featured desktop computer on the left hand side and his or her own selection on the right hand side as shown in Figure 7.

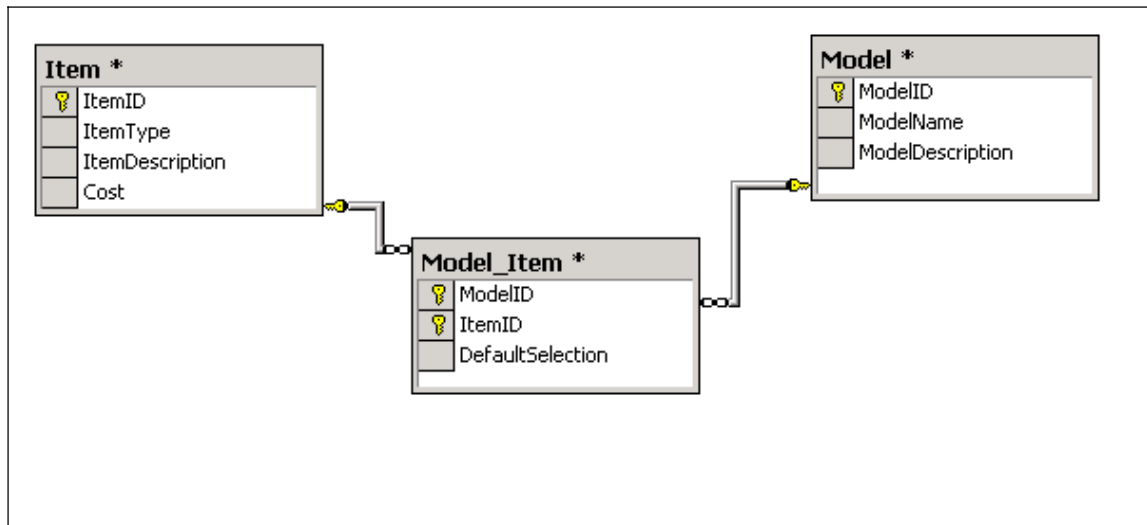


Figure 6: Database Diagram for Multi Computer Models

- In order to provide feedback to a customer, the students were asked to implement the Session concept in ASP 2.0 to allow the selected choices of the customer and the price information to be made available to other pages including the 'Confirmation' page. Figure 8 presents the confirmation page. A customer could also reconfigure his/her choice by using 'Edit Selection' button on the confirmation page. Appendix 3 presents the code provided by the students for using Session variables to maintain the selected information.
- The students felt strongly that the website could benefit from a consistent look and feel. Therefore, a Master Page was created to give the site a consistent look and feel throughout and all Web pages in the project were designed using the Master page.

One student raised an interesting question about how to keep the project cost within the budget in a real business

environment during the final round of the project's development. The concern was that the prototype approach which involved iterative steps for improvement would go on and on and that the project's cost would keep increasing and might well exceed the budget. This led to a discussion in the class which helped students realize the importance of understanding customer requirements and effective communication between designers and users. Joint designer-user accountability would be the key success factor to move to the prototype to the next level at the optimal time and to control the project cost.

3. THE HANDS-ON ENVIRONMENT

All lab computers used in this class had Windows XP and the Professional edition of Visual Studio 2005 installed. In addition, a dedicated Web server with Windows 2003 Enterprise edition was used for the class and implementation of the students' projects. This server was also equipped with

Please Customize Your Desktop: [Select My Hard Drive](#)

Base Price: \$ 605.00

Pentium D Processor 820 with Dual Core Technology (2.8GHz, 800FSB)

- 512MB Dual Channel DDR2 SDRAM at 533MHz (2x256M)

DataSafe 160GB (includes main hard drive plus a hidden reserve hard drive)

- 512MB Dual Channel DDR2 SDRAM at 533MHz (2x256M)

Single Drive: 48x CD-RW/DVD-ROM Combo Drive

- Drive

FREE UPGRADE!! 17 inch E176FP Analog Flat Panel

- Panel

Genuine Window XP Media Center Edition 2005

- Media Center Edition 2005

New Price: \$605.00





Your Customized Configuration:

- Pentium D Processor 820 with Dual Core Technology (2.8GHz, 800FSB)
- 512MB Dual Channel DDR2 SDRAM at 533MHz (2x256M)

Price: Your Selection:

\$155.00 DataSafe 160GB (includes main hard drive plus a hidden reserve hard drive)

[Help me Choose](#)

- 80GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache [subtract \$-119]
- 160GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache [subtract \$-92]
- 250GB Serial ATA 3 Gb/s Hard Drive (7200RPM) w/8MB cache [subtract \$-41]
- DataSafe 160GB (includes main hard drive plus a hidden reserve hard drive)
- DataSafe 250GB (includes main hard drive plus a hidden reserve hard drive) [add \$10]

Figure 7: Web Page for the Customization with the Featured Solution

DELL™

Order Confirmation Page

Please confirm the order information below:

Processor:	Pentium D Processor 820 with Dual Core Technology (2.8GHz, 800FSB)
Memory:	512MB Dual Channel DDR2 SDRAM at 533MHz (2x256M)
Operating System:	Genuine Window XP Media Center Edition 2005
Hard Drive:	DataSafe 160GB (includes main hard drive plus a hidden reserve hard drive)
DVD Drive:	Single Drive: 48x CD-RW/DVD-ROM Combo Drive
Monitor:	FREE UPGRADE!! 17 inch E176FP Analog Flat Panel
Storage:	
Final Price:	\$605.00

Figure 8: The Confirmation Page

Visual Studio 2005 and Microsoft SQL Server 2000. The students were asked to develop their ASP.Net applications on the lab computers. After successfully testing the applications on the lab computers, the students copied and published the applications to the remote Web server.

4. CONCLUSIONS

The project was a very good experience for the students. Many were excited that they had developed a project that was similar to Dell's website and that they were able to use ASP.Net easily. Most importantly, the students felt that the several rounds of the project activities helped them better understand EC Web application design from both the technical and user perspectives in a real business environment. For example, several students commented that "the three rounds design really helped re-learn old assignments to aid new assignments easily. It was very good use of a practical application." As a result of this project, the following are some suggestions for other faculty to teach a similar project in their EC design class:

- The students were given the opportunity to look at Dell's website before designing their project. The next class should be asked to evaluate at least three different websites such as Dell, HP, and IBM that allow customization of a computer purchase. In this way students will observe alternative designs for customization. It is hoped that this will allow students to realize that the project they are creating is similar to many websites, not just Dell's.
- The students were asked to evaluate Dell's website and to produce a website design which was similar. This type of activity can be classified as Reverse Engineering since it is the process of examining how software works and drawing useful conclusions. Reverse Engineering does play a vital role in the legitimate process of software development and it is particularly helpful for students who would like to learn project development skills in a living classroom. Adding a short lecture or discussion on the ethical issues involved in reengineering a website would be very beneficial.
- Some students had a difficult time designing their database and handling the normalization issues when they modified the database to handle multiple computer systems. Adding a short lecture on database design, normalization, and implementation would be beneficial to students to understand how the data model could support the site development.
- Students should be challenged to further explore database development to support the website application. The database tables could be expanded to include Department (Desktop/Notebook), Type (Home/Business), Model (XPS/Dimension), Item, and Model_Item to enrich to the computer systems offered through the Web application design.
- It would be very helpful to ask students to extend the project design to incorporate a shipping page and an automated email confirmation. These additional features would give students a sense of the completeness of the project.
- Prerequisites for students should include knowledge of programming language, HTML, and database concepts.

Students can learn quickly and develop the site in more depth if they have these skills.

5. ACKNOWLEDGEMENTS

The authors would like to thank the reviewers' and the assistant editor's constructive comments and suggestions to help improve the quality of this paper. The authors are willing to share the project and lab assignments upon request.

6. REFERENCE

- Hart, C., Kauffman, J., Sussman, D., and Ullman, C. (2006). *Beginning ASP.Net 2.0*, Wiley Publishing, Inc., Indianapolis, IN 46256.
- Landry, B.J., Griffith, R., and Hartman, S. (2006). Measuring student perceptions of blackboard using the technology acceptance model, *Decision Sciences Journal of Innovative Education*, 4(1), pp. 87-99.
- Liu, C. (2006). Web survey design in ASP.Net 2.0: a simple task with one line of code. *Journal of Information Systems Education*, 18(1), pp. 11-14.
- Lu, J., Yu, C.S., and Liu, C. (2003). Learning style, learning pattern, and learning performance in a WebCT-based MIS course, *Information & Management*, 40(6), pp. 497-507.

AUTHOR BIOGRAPHIES

Chang Liu, DBA, is an Associate Professor of MIS at Northern Illinois University. He received his doctor of business administration from Mississippi State University in 1997. He teaches database and electronic commerce courses and received MIS Graduate Teaching Awards in 2002, 2003, and 2005. His research works published at *Information & Management*, *International Journal of Electronic Commerce and Business Media*, *Journal of Global Information Management*, *Journal of Internet Research*, *Journal of Computer Information Systems*, *Mid-American Journal of Business*, *International Journal of Mobile Communications*, *Journal of International Technology and Information Management*, and *Journal of Informatics Education Research*.



Brian Mackie is an Associate Professor of MIS at Northern Illinois University. He received his doctor of business administration from the University of Iowa in 1999. He teaches database, operating systems and Networking. His research interests include online collaboration, Electronic commerce and Database Systems.



Appendix 1: The First Round of Project Solution for the “Update” Button

```
Private Sub btnPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrice.Click
    /* This sub-routine calculates and displays the total price when you press the “update button” */

    Dim Proc, Ram, Hd, Cd, Window, Sd, Mon As Integer
    Dim li As ListItem
    Dim sumList As Integer = 0

    Proc = Processor.SelectedItem.Value
    Ram = Memory.SelectedItem.Value
    Hd = HardDrive.SelectedItem.Value
    Cd = CDDrive.SelectedItem.Value
    Window = OS.SelectedItem.Value
    Mon = Monitor.SelectedItem.Value

    If Storage.SelectedIndex <> -1 Then
        For Each li In Storage.Items
            If li.Selected Then
                sumList += li.Value
            End If
        Next
    End If

    NewPrice.Text = Format(673 + Proc + Ram + Hd + Cd + Window + Mon + sumList, "currency")

End Sub
```

Appendix 2: The Second Round of Project Solution for Price Updating

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    If Not IsPostBack Then
        Processor.DataBind()
        Memory.DataBind()
        HardDrive.DataBind()
        DVD.DataBind()
        Screen.DataBind()
        OS.DataBind()

        Call Position(Processor)
        Call Position(Memory)
        Call Position(HardDrive)
        Call Position(DVD)
        Call Position(Screen)
        Call Position(OS)
        Call calculate()
    Else
        Call calculate()
        /* will recalculate the price if an item has been changed */
    End If
End Sub

Private Sub Position(ByVal x)
    Dim y = x.id
    Dim selection As Integer
    Dim itemNum As Integer = x.items.count
    Dim index As Integer = 0
```



```
Dim cnn As New SqlConnection(ConfigurationManager.ConnectionStrings("Case_DellConnectionString1").ConnectionString)
Dim mycommand As New SqlCommand
mycommand.Connection = cnn
mycommand.CommandText = "select * from item_E510 where itemType='" & y & "'"

cnn.Open()
Dim myReader As SqlDataReader
myReader = mycommand.ExecuteReader

Dim itemArray1(itemNum) As Integer ' for Cost
Dim itemArray2(itemNum) As String ' for ItemDescription
Dim itemArray3(itemNum) As Boolean ' for DefaultSelection

While myReader.Read
itemArray1(index) = myReader("price")
itemArray2(index) = myReader("itemContent")
itemArray3(index) = myReader("default_Value")
If itemArray3(index) = True And IsPostBack = False Then
selection = index
x.items(selection).selected = True
Else
selection = x.selectedIndex
End If
index += 1
End While

For index = 0 To itemNum - 1
Dim i = itemArray1(index) - itemArray1(selection)
If selection <> index Then
If i > 0 Then
itemArray2(index) += " [add $" + Format(CSng(i), "#.") + "]"
/* here is where the amount additional for the unselected item would be displayed */
Else
itemArray2(index) += " [subtract $" + Format(CSng(i), "#.") + "]"
/* here is where the amount less for the unselected item would be displayed */
End If
End If
x.Items(index).Text = itemArray2(index)
x.items(selection).Text = itemArray2(selection) + "[included in price]"
/* here is where the selected item has included in price displayed */
Next
End Sub

Private Sub calculate()
/* here is where amount of each item is summed together to give the total cost */

Dim pro, ram, hd, cd, moni, window, ChkSum, total As Integer
Dim li As ListItem
pro = Processor.SelectedValue
ram = Memory.SelectedValue
hd = HardDrive.SelectedValue
cd = DVD.SelectedValue
moni = Screen.SelectedValue
window = OS.SelectedValue

For Each li In Storage.Items
If li.Selected Then
ChkSum += li.Value
End If
```

```
Next
total = pro + ram + hd + cd + moni + window + ChkSum
New_Price.Text = Format(total, "currency")
    /* here is where the total amount for the current selected configuration is displayed */
If IsPostBack = False Then
Base_Price.Text = Format(total, "currency")
New_Price.Text = Base_Price.Text
    /* here is where the total amount for the base configuration is displayed */
End If
End Sub
```

Appendix 3: Using Session to Maintain User Selection Information

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    /* This sub-routine retrieves the selected components from the session variables and displays to the screen control
    labels, i.e., lblProcessor.Text would contain the selected processor, this would be displayed on the screen */

lblProcessor.Text = Session("selected_Processor")
lblOS.Text = Session("selected_OS")
lblMemory.Text = Session("selected_Memory")
lblHardDrive.Text = Session("selected_HardDrive")
lblDVD.Text = Session("selected_DVDDrive")
lblMonitor.Text = Session("selected_Monitor")
lblStorage.Text = Session("selected_Storage")
lblPrice.Text = Session("Price")
End Sub
```



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2008 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096