**UNIVERSITY OF THE PACIFIC**
**COURSE APPROVAL FORM**
**REVISION**

Please fill in all information. Required signatures are on page 2 of this form. Please return to: Academic Affairs Committee, Office of the Provost, Anderson Hall, 2nd Floor.

<table>
<thead>
<tr>
<th>Contact Person: Albert Huang</th>
<th>Phone: 946-7494</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: November 10, 2006</td>
<td>School or College: BUS</td>
</tr>
<tr>
<td>Proposed Course #: BUSI 139</td>
<td>Department: N/A</td>
</tr>
<tr>
<td>Title: Electronic Commerce Project</td>
<td></td>
</tr>
<tr>
<td>Proposed Prerequisites: BUSI-100 and Junior standing</td>
<td></td>
</tr>
<tr>
<td>Proposed Units: 4</td>
<td>Enrollment/Expected Enrollment: 15</td>
</tr>
<tr>
<td>Existing Course Title: Electronic Commerce Project</td>
<td></td>
</tr>
<tr>
<td>Existing Course #: BUSI 139</td>
<td>Grade Option: Letter Grade</td>
</tr>
<tr>
<td>Existing Unit Value: 4</td>
<td>Existing Prerequisites: BUSI 136, 137, 138 with a “C” or better or permission of the instructor; junior standing.</td>
</tr>
</tbody>
</table>

Revised catalog description (attach additional sheet if necessary). Attach a syllabus:
No change

Describe the proposed changes and provide a rationale (attach additional sheet if necessary). Drop BUSI 136, 137, 138 from the prerequisite list. They are no longer needed. See attached.

If approved, when will this be implemented? Fall [x] Spring [ ] Year 2007

What is the anticipated impact on resources (e.g., Faculty, funds, library materials, etc.)? None

Describe any special facilities, furnishings, or technical needs. List software needs, if any. None

---

**U.O.P.**
**JAN 16 2007**
**REGISTRAR**
# APPROVAL PROCESS

Please obtain all signatures before submitting to Academic Affairs Committee. Acquire signatures in the order in which they are listed below.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action Description</th>
<th>Approved by</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Action by department requesting addition/change:</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Action by the Curriculum Committee of the School/College:</td>
<td>Suzanne B. Webster</td>
<td>1/15/07</td>
</tr>
<tr>
<td>3.</td>
<td>Action by the Dean of the School/College:</td>
<td>Ray S.</td>
<td>1/15/07</td>
</tr>
<tr>
<td>4.</td>
<td>Action by the Dean of the Library:</td>
<td>Ray S.</td>
<td>1/15/07</td>
</tr>
<tr>
<td>5.</td>
<td>Action by the Director of Educational Technology Services (if computer lab, software needed):</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Action by the Registrar:</td>
<td>Cindi M.</td>
<td>1-17-07</td>
</tr>
<tr>
<td>7.</td>
<td>Action by the General Education Committee (as appropriate):</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Action by the Graduate Studies Committee (as appropriate):</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Action by the Academic Affairs Committee:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After approval by the Academic Affairs Committee, information regarding new, revised, or deleted programs and courses is sent to the Registrar for listing in or modifying the catalog.

Form revised 9/4/03
Rationale to modify prerequisites:

The primary tool used in the course is Microsoft Visual Studio. Due to recently improvement and upgrade of the tool, we are able to drop the prerequisites to provide more flexibility in course selection for students.
Jerry Post

BUSI 139: E-commerce Project

Fall 2006
MWF 11:00
E-mail: Jerry@JerryPost.com

Course Objectives

- To build interactive database-driven websites.
- To understand the design issues in building business websites.
- To understand the issues in hosting websites and selecting hosts.
- To understand security issues and procedures in securing websites.

Course Description

Designing and building interactive database-driven websites for e-business.

Software Used to Support Class

Microsoft Visual Studio, Oracle

Required Materials

Almost any HTML book or reference. Other notes will be provided in class.

Projects and Assignments

Fundamental cases and exercises are detailed in the textbook. The specific assignments are listed in the following table.
Instructor

Jerry Post

Email: Jerry@JerryPost.com

Phone: 209-946-2627

Office Hours: MW: 1-4; Th: 9:30-11:30

Attendance

Attendance is the responsibility of the student. Historically, students who miss several classes tend to receive one grade lower than other students.

I will give "make-up" exams only in the case of emergencies. The student is responsible for notifying me before the scheduled test.

Assignments

Assignments are due at class time on the specified date. Late assignments might not be accepted and can be penalized 10% per day.

Cheating

If I find assignments that are copies of each other, all students involved will receive a zero for the assignment.
Student Web Site Policies

Students in the Web class can use their accounts to create student projects. Some conditions and policies apply to these accounts. Read these policies carefully. Use of this site indicates your agreement with all of the policies. Violations of the policies could be treated as an honor code violation.

1. The accounts will be used only for student projects.
2. Students and the sites may not represent the university or any other organization. The main start page should include the student name and a notice that the site is a student project.
3. The site cannot be used for commercial or monetary transactions (except limited and controlled tests for development purposes).
4. Students should avoid material that might offend members of the community. Remember that all university student policies apply.
5. Students may not interfere with or use material from other student or faculty sites.
6. Students are responsible for all content on their site, and must adhere to all copyright, patent, and other legal restraints. All material should be the work of the student. Material that was not originally created by the student should have written copyright permissions.
7. Students should remember that faculty reserve the right to show any websites and content to prospective employers.
8. The server may be shut down at random times or may fail, or the accounts may be removed at any time. Students are responsible for backing up their own work (somewhere other than the server).
9. The server is the property of Jerry Post and students agree that he has full control over the server and all accounts.
Web Class Assignments

Each assignment must be submitted separately, must be completely your own work, and must be identified with your name and the appropriate assignment number (e.g., HTML #1). Assignments submitted after the deadline will lose 10% of the points for each day they are late. Assignments and projects are submitted by adding them to your Web site.

HTML Assignments

20 points each.

1. A page on the server with your name and a list of all assignments. Note, this assignment is designed to test the system and the login. It must be submitted before September 15 to receive any credit. Each assignment will be a link to assignments that you complete. Initially, build at least one of the links to a temporary/new page.

2. Two pages based on one style sheet with fonts specified for at least two levels of headings, and left/right margin settings for paragraphs to indent them by about one inch. Also include a class definition called CompanyName for slight bold, italic font. Use sample text to illustrate the styles.

3. A page with a set of images used as a menu: Clicking on each image brings you to a new page.

4. A small table of sales and cost data for a company. Include columns for Sales 2000 and 2001 and Costs 2000 and 2001. Use five rows of data corresponding to five different geographic regions (West, South, etc). The Column labels should use a column span list Sales over the top of two columns and Cost over the top of the next two columns.

5. A page with a typical layout with a header section at the top and at least two columns in the main section.

Client-Side Programming

20 points each.

1. Create a simple image menu with at least 3 options. Add a second roll-over image to highlight menu choices as they are selected.

2. Create a small form with a text box and a button. Write script so that the user can enter a color name or number. When the button is clicked, the script changes the background of the form to the specified color.

3. Create a small form with at least three text boxes, including one for an e-mail address. Add script to set the focus to the first box. Add script to ensure that the e-mail address contains at least an @ sign and a dot.

4. Create a page with a button. When the button is clicked it executes your code to open a second window that is of fixed size and limited controls. The page in the new window should contain a Close button that closes the new browser window.

5. Create a small form and write code to determine if any item has been changed. When the user closes the form use a popup confirmation box to ask if the changes should be saved. If they choose to save the data, send them to a different page.

Server-Side Programming

20 points each.

1. Create a simple online calculator. The user will enter numbers in two boxes, then click one of four buttons (add, subtract, multiply, divide) to get the result.
2. Create a form to order a pizza using check boxes for toppings, radio buttons for crust, and a select box to choose a city. When the order is submitted, display a description of the pizza in a box.

3. Create a form with Name, Age and State (feel free to include more data). Write code to evaluate the form and return a message. Make up different messages for the following conditions: If the state is California and they are over 18. If the state is Nevada and they are under 18. If the state is Nevada, use different messages for each of the following: under 10, between 11 and 15, between 16 and 20, over 21.

4. Create a form that has a starting value, and ending value, and a result box. Write a loop that adds all of the integers between the starting and ending number. For example, 1+2+3+4+5 = 15.

5. Create a form that has boxes to enter a name and an age. Write code that alters and then displays a hidden message label based on the user's age. For example, for users between 21 and 25, make the text red and larger.

Database

20 points each.

1. Create a database with an Employee table. Create a main form to edit data for one employee. Enter sample data for 20 employees.
2. Write SQL statements to perform the following tasks:
   a. List all employees.
   b. Insert an employee.
   c. Change a last name for a given Employee ID value.
   d. Update a phone number using optimistic concurrency (where you know the original value and the new value).
   e. Delete an employee.
3. Create a form that enables a manager to enter a salary value. Then return a list of all the employees making more than that amount and their total salary.
4. Create a grid form that an administrator could use to edit data for multiple employees.
5. Add a password and username to the Employee table. Create forms and code for the employee to login, verify the username, password, and then change his or her address and phone number.

Research

20 points each. Best answer gets 5 extra points.

1. Find a Web color picker and find a color wheel.
2. Find a detailed HTML reference site.
3. Find a rollover/drop-down menu system and identify which browsers it works with.
4. Find a site that sells security certificates and identify the steps needed to (a) obtain the certificate, (b) install it on the server, and (c) activate it in an application.
5. Find a company to host an e-Commerce Web site using ASP.NET with an expected client load of between 500 and 100 hits per day, some with large data transfers (20?100 MB downloads).