Lab 3: HTML Concepts

In this lab, you will add functionality to the Web site you created in the last lab.

Objectives:

- To write an HTML form
- To use JavaScript to validate form data on the client
- To write a Cascading Style Sheet

Part 1: Writing an HTML Form

Steps:

_1. Ask your instructor for the following information:

Lab Installation directory: _____

If your computer has the standard lab setup, the directories will be something like:

```
Lab Installation Directory:       c:\internetclass
```

_2. If it's not running, start WSAD by choosing **Application Developer 5.1** from the **IBM WebSphere Studio** entry on the Windows Start menu.

When prompted for the *Workspace Location*, enter the **{Lab Installation Directory}\workspace**, for example:

```
c:\internetclass\workspace
```

_3. Your first job is to create a new *project* for your lab.

a. Choose *File - New - Dynamic Web Project*. On the first page of the wizard, enter:

```
Project name:                Lab03Web
Configure advanced options:  checked
```

Then press Next.

b. On the *J2EE Settings Page*, note that the *Context root* is *Lab03Web* -- this will act as a prefix on the URLs for your Web pages.

Then enter:

Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

```
EAR Project:  Lab03EAR
```

Then press Next.

c. On the *Features Page*, uncheck the *Default style sheet (CSS)* option (you will create your style sheet in a later lab), then press Next.

d. On the *Select Page Template* page, note that we are not using a page template for our Web site, then press Finish.

e. Expand the *Lab03Web* entry in the Project Navigator so you can see the *WebContent* folder.

_4. To save time, let's copy files from the last lab. If you didn't finish the last lab, contact the instructor about how to copy from the last lab's solution.

Follow these steps to copy:

a. Expand the *Lab02Web* and nested *WebContent* folders.

b. In *Lab02Web/WebContent*, right-click on *east.html* and choose **Copy**. Then right-click on *Lab03Web/WebContent* and choose **Paste**. You should see the file in the new Project. WSAD will display a "broken-link" warning that you will fix in a moment.

Repeat the Copy-Paste for the rest of the HTML files and the GIF file. You should have no broken-link warnings at this point.

_5. Edit *index.html* by double-clicking on its entry in the Project Navigator.

Above the "Locations" heading, inside of new paragraph, add a hyperlink to a page named *orderform.html* with the text **Order a Pizza Now!**. WSAD will show a "broken-link" warning that you will fix in a moment.

_6. Using the same technique as in the last lab, create a new page named **orderform.html** in the *WebContent* folder, then delete all of the WSAD-provided content.

You should have no "broken-link" warnings at this point.

Write the boilerplate HTML and the common content for the new page:

- The page title should be *Online Order Form*
- The page should display the parlor's logo
- The page should have a level-three heading that displays the page title
- The page should have the same horizontal rule and copyright notice as your first page

_7. The new page will contain an HTML form betwen the level-three heading and the horizontal rule. Start by defining the *form* element with an *action* of **OrderPizza** using the *post* HTTP method. Be sure to write the closing tag for the form.

Note: At this point, you will have a broken-link warning that you can ignore for now.

_8. Now you can complete the form definition. Follow these steps to insert user-input widgets within the form:

Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

a. Inside of a paragraph (within the form), write the text **Name** and define a text-input widget with the *name* set to **name**.

   Use preview to see if the "Name" text and the text-input widget look like you expect.

b. Repeat the above step to define two more text-input widgets labeled **Phone** and **Address** using appropriate *names* (we recommend using all lower-case for the *names*).

   Note: Don't worry too much if the text-input widgets don't line up nicely. The goal here is create a form, not to get it to look pretty.

c. Inside of another paragraph, define a *select* widget with the label **Store Location:** and name **location**. The *option* values should be the four store locations (East, West, North and South).

d. Inside of a paragraph, write the text **Pizza size:**

   Inside of another paragraph, define a *radio* widget with *name* set to **size** and *value* set to **small** and then write the text **Small**.

   In a similar fashion, define two more radio buttons for medium and large pizzas. The *Large* entry should be pre-checked.

e. Finally, in yet another paragraph, define a Submit button with *value* set to **Order Now!**

f. Use the Preview mode to examine your form. It should look something like:

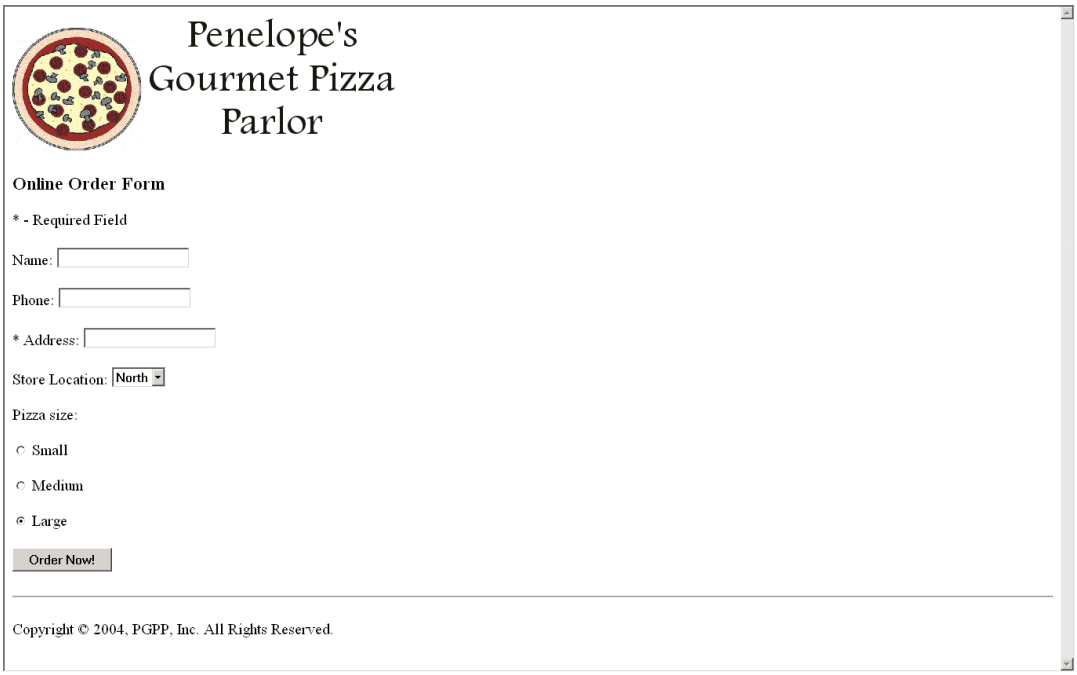


Figure 1: Screen Shot

You can enter data and choose a location and size, but don't submit the form (you don't have a program to handle the submisssion).

Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

In particular, note that when you select a radio button, the previous selection should de-select.

Part 2: Working with JavaScript

In this part, you will write some simple JavaScript to ensure that the user fills in required fields in the form.

Steps:

_1. Edit *orderform.html* and add an asterisk in front of the *Address* text-field label. Then add a new parapgraph above the form that reads:

```
* - Required Field
```

_2. Modify the *form* start-tag so that the form has a *name* of **OrderForm** and a *onsubmit* attribute of **return validateOrderForm()**.

_3. In the *head* section of the page, write JavaScript that checks to see if the *address* field has any text in it and shows an alert if it's empty.

Use the sample JavaScript program in the course notes for guidance, but be sure to change the names of variables, etc. as appropriate.

_4. To test your JavaScript, preview the page and press the Submit button without entering an address -- you should see an alert message box. Note that this checking should work even though we don't have a server program to handle the page: that's because the JavaScript program runs within the client browser.

Part 3: Using Cascading Style Sheets

In this part, you will write a cascading style sheet that governs the overall appearance of your Web site.

Steps:

_1. Right-click on the *WebContent* folder and choose **New - CSS File**. Name the file **general.css**.

WSAD will create the new file and display the CSS editor, which consists of a preview pane and an editing pane into which you can type CSS selectors.

_2. In the right-hand pane, write a CSS rule for the *h3* tag with the following properties:

```
font-family:            sans-serif;
color:                  olive;
```

Note how the preview pane shows your style.

_3. Write additional CSS rules for the following tags:

```
h4
```


Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

```
----
font-family:            sans-serif;
color:                  navy;
border-bottom:          medium solid navy;

 p
---
font-family:            Verdana,Arial,Helvetica,sans-serif;

p.footer
-------
font-family:            Verdana,Arial,Helvetica,sans-serif;
font-size:              xx-small;
text-align:             center;
```

_4. Next, update your **index.html** so it references the external CSS file. Use the preview pane of the HTML editor to examine the results.

   Then update the copyright paragraph so it uses the *p.footer* style.

_5. Repeat the above for the rest of the pages on your Web site.

_6. As in the last lab, run the *index.html* file on the server and test your Web site (everything but submitting the filled-in form).

_7. Change the stylesheet so that the *h3* text is red, rather than olive. Then retest. How easy was it to modify the look of an entire Web site?

Optional:

Optional labs are for students that wish to explore topics with less "hand-holding" than the normal labs. If you decide to do optional labs, you are expected to finish them with minimal help!

_1. Use a borderless table (border="0") to organize the display of your form so that the text-input widgets line up nicely.

_2. Update your JavaScript so that it checks for the Name and Phone text-input widgets as required. The alert box should display all of the missing fields (i.e. there should only be one alert() call in your JavaScript).

Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

Copyright © Descriptor Systems 2004. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.