**Start creating the technology journal subscription page for Central Valley Technology**

1. Open subscription.html.
2. Locate the [Add code here] in the document body and replace it with the following headings:

<h1> Technology Journal Subscri ption Form </h1>

<h2> Customer Information </h2>

1. Add the following after the <h2> tag to create the form section.

<form encType="application/x-www-form-urlencoded" method="get" action="FormProcessor.html">

<p><input value="Subscribe" type="submit">&nbsp;<input value="Reset" type="reset"></p>

</form>

1. Save the document.

**To add text <input> elements to the subscription form.**

1. Open subscription.html.
2. Above the paragraph element containing the submit and reset buttons, add the following table to contain the billing and shipping information text boxes. The table will consist of a single row containing 2 cells.

<table border=”0”>

</table>

1. Within the table, add the following text <input> elements, which will be used to gather a customer’s billing information and shipping information. The left cell contains the billing information, and the right cell contains the shipping information.

<tr> <td valign="top">

<h3>Billing Information</h3>

<p> Name<br /><input name="name\_billing" size="56" type="text"></p>

<p>Address<br /><input name="address\_billing" size="56" type="text"   
 /></p>

<p> City, State, Zip<br /><input name="city\_billing" size="34" type="text" />

<input name="state\_billing" maxlength="2" size="2" type="text" />

<input name="zip\_billing" maxlength="10" size="10" type="text"></p>

</td>

<td>&nbsp;</td>

<td valign="top">

<h3> Shipping Information</h3>

<p>Name<br /><input name="name\_shipping" size="56" type="text" /></p>

<p> Address<br /><input name="address\_shipping" size="56" type="text" /></p>

<p> City, State, Zip<br /><input name="city\_shipping" size="34" type="text" />

<input name="state\_shipping" maxlength="2" size="2" type="text" />

<input name="zip\_shipping" maxlength="5" size="10" type="text" /></p></td></tr>

1. After the closing </table> tag, add the following elements for the telephone number:

<p>Telephone </p>

<p> ( <input name="area" maxlength="3" size="3" type="text" />)

<input name="exchange" maxlength="3" size="3" type="text" />

<input name="phone" maxlength="4" size="4" type="text" /></p>

1. Save the file and view it to make sure the elements appear.

**To add a function to the subscription.html document that checks whether a value entered into the ZIP code or telephone number fields is a number:**

1. Open the subscription.html file.
2. Add the following script section to the end of the document head:

<script type="text/javascript">

/\* <![CDATA[ \*/

/\* ]]> \*/

</script>

1. Add the following checkForNumber() function to the script section in the document head. The function checks whether the argument passed to it is a number. If it is not a number, an alert message is displayed and the function returns a value of false to prevent the blur event from occurring. Otherwise, the function returns a true, allowing the blur event to occur.

function checkForNumber(fieldValue) {

var numberCheck = isNaN(fieldValue);

if (numberCheck == true) {

window.alert("You must enter a numeric value!");

return false;

}

else

return true;

}

1. Next, add the following onblur event handler to the <input> elements for the 2 Zip code fields and the 3 telephone number fields. Notice that the function is passed a value of this.value, which uses the *this* reference to refer to the value property of the current form element.

onblur="return checkForNumber(this.value);"

1. Save the file and test the validation.

**To add a password <input> element to the subscription.html file:**

1. Open the subscription.html file.
2. Above the paragraph element containing the submit and reset buttons, add the following lines for the user name field and password <input> element, which prompts users for a password:

<p>User name<br /> <input type="text" name="userName" size="50" /></p>

<p>Password<br /><input type="password" name="password" size="50" /></p>

1. Save the file and test that the password field shows asterisks or bullets.

**To add a password confirmation field to the subscription.html file:**

1. Open the subscription .html file.
2. Above the paragraph element containing the Submit and Rest buttons at the end of the form, add the following lines for a password confirmation field. The <input> field includes an onblur() event handler to call the confirmPassword() function.

<p>Confirm password<br /><input type="password" name="password\_confirm" size="50" onblur="confirmPassword();" /></p>

1. Add the following confirmPassword() function to the end of the script section. The function compares the values entered into both the password and password confirmation fields. If the values are not the same, then the function uses the focus() method to move the cursor back into the password field.

function confirmPassword() {

if (document.forms[0].password\_confirm.value != document.forms[0].password.value) {

window.alert("You did not enter the same password!");

document.forms[0].password.focus();

}

}

1. Save and test the file.

**To add radio buttons to the subscription.html file:**

1. Open subscription.html.
2. Add the following <h3> element and opening <table> element and the tables header information to the end of the form but above the paragraph element containing the submit and reset buttons.

<h3>Delivery Rates</h3>

<table border="0">

<colgroup align="left" width="100" />

<colgroup span="4" align="center" width="100" />

<tr> <th> 6 issues </th>

<th>12 issues</th>

<th>18 issues</th>

<th>24 issues </th>

</tr>

1. Next, add the following table row elements for payment options by number of issues. Notice that each radio buttons name attribute is assigned the same name of “delivery” so that the radio buttons are part of the same group.

<tr><td><input type="radio" name="delivery" value="17.50" />$17.50 </td>

<td><input type="radio" name="delivery" value="26.95" />$26.95</td>

<td><input type="radio" name="delivery" value="32.90" />$32.90 </td>

<td><input type="radio" name="delivery" value="46.00" />$46.00</td>

</tr></table>

1. Save the file and test it to make sure only 1 radio button can be selected.

**To add monthly and yearly subscription radio buttons to the subscription.html file:**

1. Open subscription.html file.
2. Add the following radio buttons for automatic renewal above the paragraphs containing the submit and reset buttons. The buttons call an onclick event handler named billAutomatically(), which you will create next.

<p><strong>Automatic Renewal</strong>:

<input type="radio" name="autoRenew" onclick="billAutomatically();" />Monthly ($8.95)

<input type="radio" name="autoRenew" onclick="billAutomatically();" />Yearly ($19.95) </p>

1. Add the following 2 functions to the end of the script section. The billAutomatically() function disables any selected radio button in the payment per number of issues group, and the billByIssue() function disables the selected radio button in the automatic renewal group.

function billAutomatically() {

for (var i = 0; i < document.forms[0].delivery.length; ++i) {

if (document.forms[0].delivery[i].checked == true) {

document.forms[0].delivery[i].checked = false;

break;

}

}

}

function billByIssue() {

for (var i = 0; i < document.forms[0].autoRenew.length; ++i) {

if (document.forms[0].autoRenew[i].checked == true) {

document.forms[0].autoRenew[i].checked = false;

break;

}

}

}

1. Add onclick event handlers to the radio buttons in the delivery rates group that call the billByIssue() function, as follows:

<td><input type="radio" name="delivery" value="17.50" onclick="billByIssue();" />$17.50</td>

<td><input type="radio" name="delivery" value="26.95" onclick="billByIssue();" />$26.95 </td>

<td><input type="radio" name="delivery" value="32.90" onclick="billByIssue();" />$32.90</td>

<td><input type="radio" name="delivery" value="46.00" onclick="billByIssue();" />$46.00 </td>

1. Save the file and test the radio buttons to make sure you can only select 1 button from either group.

**Next you will add check boxes to the subscription.html file to allow users to select the type of technology they are interested in:**

1. Open subscription.html file.
2. Above the paragraph containing the submit and reset buttons, add the following check box elements:

<p>What technologies are you interested in?</p>

<p><input type="checkbox" name="technologies" value="architecture" />Architecture<br />

<input type="checkbox" name="technologies" value="hardware" />Hardware<br />

<input type="checkbox" name="technologies" value="open\_source" />Open source<br />

<input type="checkbox" name="technologies" value="data" />Data management<br />

<input type="checkbox" name="technologies" value="windows" />Windows<br />

<input type="checkbox" name="technologies" value="mac" />Macintosh<br />

<input type="checkbox" name="technologies" value="networking" >Networking

</p>

1. Save the file and view it.

**To add a check box element to the subscription form that copies the values of the billing information fields to the shipping information fields:**

1. Open subscription.html file.
2. Add the following sameShippingInfo() function to the end of the script section. The function uses an *if… else* statement to copy the values from the billing information fields to the shipping information fields when the check box is selected. When the check box is deselected, empty strings are assigned to the shipping information fields. Notice that the conditional expression in the if statement uses the elements[] array to refer to the check box control. You will not assign a name to the check box control because you do not want its value submitted to the web server. Therefore, you must refer to it with the elements[] array.

function sameShippingInfo() {

if (document.forms[0].elements[5].checked == true) {

document.forms[0].name\_shipping.value

= document.forms[0].name\_billing.value;

document.forms[0].address\_shipping.value

= document.forms[0].address\_billing.value;

document.forms[0].city\_shipping.value

= document.forms[0].city\_billing.value;

document.forms[0].state\_shipping.value

= document.forms[0].state\_billing.value;

document.forms[0].zip\_shipping.value

= document.forms[0].zip\_billing.value;

}

else {

document.forms[0].name\_shipping.value = "";

document.forms[0].address\_shipping.value = "";

document.forms[0].city\_shipping.value = "";

document.forms[0].state\_shipping.value = "";

document.forms[0].zip\_shipping.value = "";

}

}

1. Now add the following check box element above the closing </td> tag for the table cell that contains the billing information fields.

<p><input type="checkbox" onclick="sameShippingInfo();" />

Same shipping information</p>

1. Save and test the check box.

**To add a selection list to the subscription.html file:**

1. Open subscription.html file.
2. Add the following selection list above the paragraph that contains the submit and reset buttons.

<p>What is your job title?</p>

<p>

<select name="jobTitle">

<option value="itStaff">IT staff</option>

<option value="consultant">Technical consultant</option>

<option value="integrator">Systems integrator</option>

<option value="manager">Manager</option>

<option value="director">Director</option>

<option value="vp">Vice President</option>

<option value="seniorManagement">CIO, CTO, CSO, CEO, COO, Chairman, President</option>

<option value="other">Other</option>

</select></p>

1. Save and view the drop down box. Because you did not include size and multiple attributes in the <select> element, the selection list appears as a drop-down menu.

**To add code to the subscription.html file that allows subscribers to build a selection of the technology journals to which they are currently subscribed.**

1. Open subscription.html file.
2. Add the following elements above the paragraph that contains the submit and reset buttons. The first element is a text box where users can enter the name of a journal to which they subscribe. Note that the text box does not include a name argument; this prevents the field value from being submitted along with the rest of the form data. The Add Journal button includes an onclick event handler that calls a function named addJournal(), which you will add next. In order for the document to be well formed, the <select> element must include at least one <option> element. Therefore, you add a single <option> element that displays “Enter the technologies journals you subscribe to” in the selection list. In the next section, you will add code to the addJournal() function that deletes this unnecessary element when the subscriber adds a journal to the selection list.

<p>Journal

<input type="text" size="68" /></p>

<p><input type="button" value="Add Journal" onclick="addJournal();" style="width: 120px" /> </p>

<p><select name="journals" multiple="multiple" size="10" style="width: 500px">

<option value="none">Enter the technologies journals you subscribe to</option>

</select></p>

1. Add the following addJournal() function to the end of the script section. Notice that the code uses the element[] array to refer to the text where users enter the names of journals to which they subscribe; this is necessary because the text box does not include a name argument. Also notice how the function determines where to add the new item in the options[] array. The number of items in the options[] array is retrieved using the Select objects length property and assigned to a variable named nextItem. The value assigned to the nextItem variable represents the number of elements in the array. As you will recall the length of the array is one more than the number of elements in the array, that means you can use the value to identify the next available element.

function addJournal() {

if (document.forms[0].elements[31].value == "")

window.alert("You must enter a journal name.");

else {

var journal = new Option();

journal.text = document.forms[0].elements[31].value;

journal.value = document.forms[0].elements[31].value;

nextItem = document.forms[0].journals.length;

document.forms[0].journals.options[nextItem] = journal;

document.forms[0].elements[31].value = "";

}

}

1. Save the file and try adding some journals. You will notice that the text in the box stays there even after you add a journal; we will fix that in the next section.

**To add code to the subscription.html document that deletes journal names from the select list:**

1. Open subscription.html file.
2. Add the following elements to the end of the <p> element that contains the Add Journal button. The Delete Journal button includes an onclick event handler, which calls a function named deleteJournal(). You will add this function next. To the onclick event handler in the clear List button, you will assign a statement that deletes all the items in the list.

<input style="width: 120px;" onclick="deleteJournal()" value="Delete Journal" type="button">

<input style="width: 120px;" onclick="document.forms[0].journals.options.length = 0;" value="Clear List" type="button">

1. Add the following deleteJournal() function to the end of the script section.

function deleteJournal() {

var selectedItem =

document.forms[0].journals.selectedIndex;

if (selectedItem == -1)

window.alert("You must select a journal name in the list.");

else

document.forms[0].journals.remove(selectedItem);

}

1. Finally add the following if statement to the beginning of the else statement in the addJournal () function. This statement deletes the “Enter the technologies you subscribe to” option that is displayed by default in the selection list.

if (document.forms[0].journals.options[0] && document.forms[0].journals.options[0].value == "none")

document.forms[0].journals.remove(0);

1. Save the file, and test these buttons.

**To add code to the subscription.html file that modifies journal names in the selection list:**

1. Open subscription.html file.
2. Add the following elements to the end of the <p> element that contains the other buttons that build the list of journals. The Change Journal button includes an onclick event handler which calls a function named changeJournal(). You will add this function next.

<input style="width: 120px;" onclick="changeJournal()" value="Change Journal" type="button">

1. Add the following changeJournal() function to the end of the script section.

function changeJournal() {

var selectedItem =

document.forms[0].journals.selectedIndex;

if (selectedItem == -1)

window.alert("You must select a journal name in the list.");

else {

document.forms[0].journals.options[selectedItem].value = document.forms[0].elements[31].value;

document.forms[0].journals.options[selectedItem].text

= document.forms[0].elements[31].value;

}

}

1. Save the file and try changing some journals in the journal list.

**To add onsubmit and onreset event handlers to the subscription.html file:**

1. Open subscription.html file.
2. Add the following confirmSubmit() and confirmReset() function to the end of the script section.

function confirmSubmit() {

var submitForm = window.confirm("Are you sure you want to submit the form?");

if (submitForm == true)

return true;

return false;

}

function confirmReset() {

var resetForm = window.confirm("Are you sure you want to reset the form?");

if (resetForm == true)

return true;

return false;

}

1. Before the closing bracket of the opening <form> tag, add the following onsubmit and onreset event handlers, which call the confirmSubmit() and confirmRest() functions:

onsubmit="return confirmSubmit();" onreset="return confirmReset();"

1. Save the file and test the buttons.

**To add code to the confirmSubmit() function in the subscription form that validates the text and password boxes:**

1. Open subscription.html file.
2. Replace the statements within the confirmSubmit function with the following statements, which validates the billing fields. The conditional statement uses multiple || (or) operators to validate all of the billing fields in 1 expression.

if (document.forms[0].name\_billing.value == ""

|| document.forms[0].address\_billing.value == ""

|| document.forms[0].city\_billing.value == ""

|| document.forms[0].state\_billing.value == ""

|| document.forms[0].zip\_billing.value == "") {

window.alert("You must enter your billing information.");

return false;

}

1. Add the following statements to the end of the confirmSubmit() function to validate the shipping fields.

else if (document.forms[0].name\_shipping.value == ""

|| document.forms[0].address\_shipping.value == ""

|| document.forms[0].city\_shipping.value == ""

|| document.forms[0].state\_shipping.value == ""

|| document.forms[0].zip\_shipping.value == "") {

window.alert("You must enter your shipping information.");

return false;

}

1. Add the following statements to the end of the confirmSubmit() function to validate the telephone fields.

else if (document.forms[0].area.value == ""

|| document.forms[0].exchange.value == ""

|| document.forms[0].phone.value == "") {

window.alert("You must enter your telephone number.");

return false;

}

1. Add the following statements to the end of the confirmSubmit() function to validate the user name.

else if (document.forms[0].userName.value == "") {

window.alert("You must enter a user name.");

return false;

}

1. Add the following statements to the end of the confirmSubmit() function to validate the password fields.

else if (document.forms[0].password.value == ""

|| document.forms[0].password\_confirm.value == "") {

window.alert("You must enter a password.");

return false;

}

1. Finally, add the following return statement to the end of the confirmSubmit() function.

return true;

1. Save the file, and test it.

**To add code to the confirmSubmit() function in the subscription form that validates the Delivery Rates radio buttons:**

1. Open subscription.html file.
2. The Delivery Rates radio buttons are contained within 2 groups: delivery and autoRenew. You need to ensure that a button is selected in one of the groups. First, add the following variable declaration and for statement above the return true; statement at the end of the confirmSubmit() function. The for statement checks whether a button is selected in the delivery group.

var deliverySelected = false;

for (var i = 0; i < 4; ++i) {

if (document.forms[0].delivery[i].checked == true) {

deliverySelected = true;

break;

}

}

1. Next, add the following for statement above the return true; statement at the end of the confirmSubmit() function to check whether a button is selected in the autoRenew group.

for (var j = 0; j < 2; ++j) {

if (document.forms[0].autoRenew[j].checked == true) {

deliverySelected = true;

break;

}

}

1. Finally, add the following code above the return true; statement at the end of the confirmSubmit() function to cancel the form submission if one of the Delivery Rates radio buttons is not selected.

if (deliverySelected != true) {

window.alert("You must select a delivery rate option.");

return false;

}

1. Save the file and test it. Enter data in the forms billing, shipping, telephone, user name and password fields and then click Subscribe. You should see an alert message instructing you to select a delivery option.

**To add a looping statement to the confirmSubmit() event handler function that selects each option in the journal selection list.**

1. Open subscription.html file.
2. Locate the return true; statement at the end of the conformSubmit function. Above the return true statement add the following code to select all of the journals in the selection list when the form is selected.

for (var k = 0; k < document.forms[0].journals.length; ++k) {

document.forms[0].journals.options[k].selected = true;

}

1. Save the file and test it, don’t forget to click the Subscribe button. The data you entered, including the journal names, should appear in the FormProcessor.html document.