MODULE 1: PERFORM PROGRAMMING IN HTML5 WITH JAVASCRIPT AND CSS3

Week 6 - 7

UNIT 2: Create HTML5 document using advanced techniques with JavaScript and CSS3

Most Essential Learning Competencies: *At the end of the course, you must be able to:*

1. Decode HTML forms
2. Discuss input types new attributes

Reading Activity

HTML Code for a Registration Form

Compared to static web pages, forms are slightly more complex to code in HTML. Following is a new HTML file that includes form fields and a submit button. HTML tags are in uppercase for clarity only.

```
<HTML>
<HEAD>
<TITLE>AICS-SHS ICT Symposium Registration Page </TITLE>
</HEAD>
<BODY><CENTER>
<H1>AICS-SHS ICT Symposium Registration Form</H1>
<HR>
<FORM METHOD=POST ACTION="http://www.amgraf.com/emailxml/save_emailxml-pl.cgi">
<P>Last Name:
<INPUT TYPE="text" NAME="LastName">
<P>First Name:
<INPUT TYPE="text" NAME="FirstName"><BR><BR>
<INPUT TYPE="radio" NAME="ict" value="FTA"><LABEL>First Time Attendee</LABEL>
<INPUT TYPE="radio" NAME="ict" value="RA"><LABEL>Repeat Attendee</LABEL><BR>
<P>Arriving On:
<SELECT NAME="ArrivalDay">
<OPTION value="1">Saturday</OPTION>
<OPTION value="2">Sunday</OPTION>
<OPTION value="3">Monday</OPTION>
<OPTION value="4">Tuesday</OPTION>
</SELECT>
<P>E-Mail Address:
<INPUT TYPE="text" NAME="EmailAddr">
<INPUT TYPE="submit" value="Register Me Please">
</FORM>
</CENTER></BODY>
</HTML>
```

Below is the expected output.

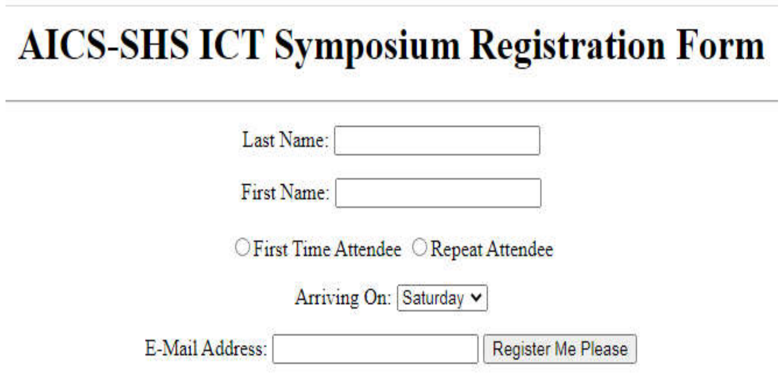

Figure 6-7.1

Decoding the HTML from the Registration Form Page

We already looked at the tags in a static HTML page. Now let's break down the additional HTML tags needed to create a fillable, submittable form.

- The <FORM METHOD=POST ACTION="http://www.amgraf.com/emailxml/save_emailxml-pl.cgi"> line is the indicator to the browser that this page contains a form. The Method attribute indicates that the data from the fill-fields will be posted to the server. This means that there will be fieldname/data value pairs (one pair for each field) transmitted to the server when the Submit button is clicked. The Action attribute indicates the URL of the program on the server that will process the form data.

- The <INPUT TYPE="text" NAME="LastName"> line is the HTML code necessary to create the first fill-field. This is the field that will capture the registrant's last name. The field is to be used for keying textual information and the field name is "LastName". A second similar input field follows for the first name.

- The next line contains <INPUT TYPE="radio" NAME="BHDT" value="FTA"><LABEL>First Time Attendee</LABEL>. This is the code to create a field type called Radio Button. On this form there are two mutually exclusive radio buttons. Only one of the buttons can be on at a time. This action is controlled by duplicating the NAME="BHDT" attribute for this field and for the next field. The value returned to the server if this field is clicked on, will be "FTA", my notation for First Timer.

- To create the dropdown list, the following code is inserted. This causes a drop down list with 4 choices to be displayed. The field name is ArrivalDay, and the value submitted to the server is "1" if the choice is Tuesday, "2" if Wednesday, etc.

```
<SELECT NAME="ArrivalDay">
<OPTION value="1">Saturday</OPTION>
<OPTION value="2">Sunday</OPTION>
<OPTION value="3">Monday</OPTION>
```

CP121

```
<OPTION value="4">Tuesday</OPTION>
</SELECT>
```

Finally, the form is submitted by clicking the Submit button drawn by the <INPUT TYPE="submit" value="Register Me Please"> code. The file ends with the closing tags for </FORM> </CENTER> </BODY> and </HTML>.

Submitting and Testing the HTML Form

A connection to the Internet is all that is required to submit the form fill data. The fieldname/data value pairs (one pair for each field) will be transmitted to the forms handler on the web server when the Submit button is clicked. In this example, the form data will then be emailed to the address entered into the sample form. A more sophisticated system would also store the information into database tables, as shown in the following diagram.

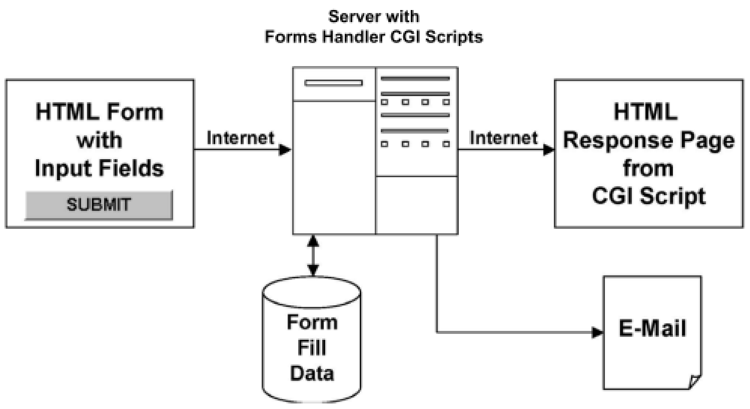
Server with
Forms Handler CGI Scripts
HTML Form with Input Fields
SUBMIT
Internet
Internet
HTML Response Page from CGI Script
Form Fill Data
E-Mail

Figure 6-7.2

HTML Forms Interface to Web Server

There is one other significant technical issue to explain. Since fillable forms are most valuable when the field content can be captured on a web server, a complete HTML forms design solution must manage the web server interfacing. The basic web server functions allow forms to be opened, filled, submitted, and saved. These server functions are usually programmed using web server scripting languages such as CGI/Perl or ASP.

A web server script is a command list that is executed by an Internet web server to direct the page management processes. The forms handler scripts provide the critical link between submittable forms and the server database management system. Some of the most common web server scripts are listed below:

- Web Server Scripts
- Forms Handler Scripts
- Open new unfilled form
- Open filled form
- Open flattened form

- Query and retrieve data from tables
- Save submitted form data

For pre-populated forms, the open filled form script retrieves the appropriate data record and fills the necessary form fields before serving the form to the client. For submittable forms, the save submitted form data script stores the field data into a table record. For dynamic database view forms, the query and retrieve field data sends a query when the client keys in a data value (i.e. account number) and returns data values to repopulate multiple form fields. This script can also retrieve a collection of values to populate a drop-down list. For field-flattened forms, the open flattened form script retrieves the appropriate data record and replaces fill fields with inline text before serving the form to the client. Another class of scripts assists users and administrators in organizing their forms and building forms-oriented workflows:

- Client Support Scripts
  - Search for records
  - List records
  - E-mail records

- Administrator Support Scripts
  - Create/Drop data tables
  - Examine data tables
  - Delete records
  - Import/Export record data as XML

- User Access Control Scripts
  - Manage Login Password/ID
  - User Profiles
  - Administrative (Who has access to Which forms)

- Workflow Processing Scripts
  - Approvals
  - Tracking
  - Reporting
  - Connectivity to other Business Systems

CP121

The Components of an Online I-forms System

An online I-forms system has essentially the same architecture as an e-commerce system, without the shopping cart and payment processing modules. Instead, the I-forms system may include expanded workflows for digital signatures, form approvals, and tracking.

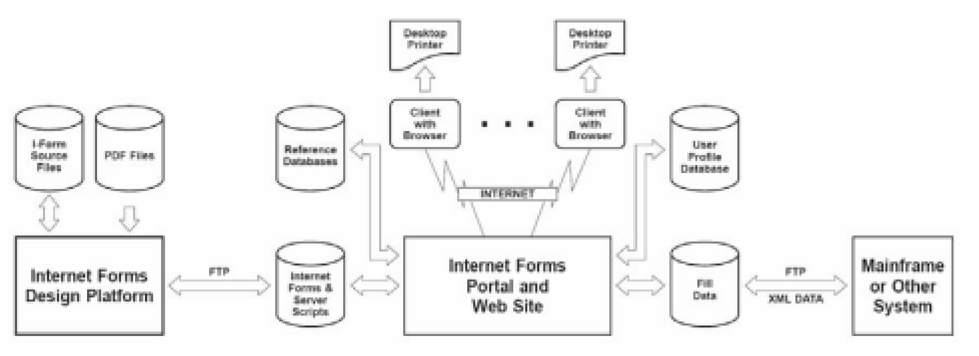

Diagram of a Typical Online Internet Forms System

Figure 6-7.3

The above architecture is ideal for centralized forms and database management on a large scale, using the Internet as the communication medium. This method is typically handled by utilizing the following technologies:

- Server-side Relational Database Management System (RDBMS)
- Web Server to Collect and Retrieve Form Field Data
- Internet Forms Repository
- Windows PC's with the Microsoft Internet Explorer (or compatible) Browser

The required software components are:

- Relational Database Management System (RDBMS)
- Oracle
- SQL Server 2000
- MySQL
- DB2
- Forms Repository
- Save by Classification and Category
- Save by Data Format (HTML, PDF, Word, etc.)
- Maintain Form Versions
- Web Server
- IIS
- Apache

Self-Check

Quiz 6-7 .1

Instructions: Write your answer on the Answer Sheet (AS) provided in this module (1-point each).

1-7. List of some common web scripts.
8-11. List of typical method of an ideal architecture for centralized forms.
12-20. List of database management systems.

Laboratory Activity

Activity 4-5.1

Additional HTML5 New Attributes

A. Open **Brackets** (code for the web) text editor or any text editor of your choice.

B. Copy-paste each html codes below and save all files respectively the folder named **HTML5 Web Forms** in your desktop and open the file using your available web browser to see the output.

1. **Filename: autocomplete.html**
Modern browsers support autocomplete, by which previous values against input elements with the same attribute name appear with IntelliSense-like behavior. By default, autocomplete on an input element inherits from the form's attribute, which defaults to "on." By setting the autocomplete attribute to off, the browser no longer retains previous values, thereby not exposing sensitive information.

```
<form action="#">

  <input type="text" name="firstName" autocomplete="off"

    placeholder="First name" /><br />

  <input type="text" name="lastName"

    placeholder="Last name" /><br />

  <input type="submit" value="Submit" />

</form>
```

2. **Filename: minmax.html**
   Used in conjunction with the number and date/datetime/time input type elements, min and max provide maximum and minimum validation of the data they're associated with. Note: These attributes don't have any effect when used with the text element type.

```
<form action="#">

<input type="number" name="age"

placeholder="Age" min="0" max="120" />

</form>
```

3. **Filename: step.html**
   Like min/max, this attribute is intended for the number, date/datetime/time type input elements. Browsers that support this attribute generally provide a spinner control to the end of the input textbox and allow the value to be incremented/decremented by the step value.

```
<form action="#">

  <input type="number" name="generation"

    placeholder="Age"

    min="0" max="120" step="10" /><br />

  <input type="submit" value="Submit" />

</form>
```

4. **Filename: required.html**
   Forces an input element to be filled in before submitting a form.

```
<form action="#">

  <input type="text" name="firstName" required

    placeholder="First name" /><br />

  <input type="text" name="lastName" required

    placeholder="Last name" /><br />
  <input type="submit" value="Submit" />

</form>
```

5. **Filename: autofocus.html**
   Placed uniquely on an input element, the autofocus attribute identifies which input element to place the focus on by default when the form loads.

```
<form action="#">

  <input type="text" name="firstName"

    placeholder="First name" /><br />

  <input type="text" name="lastName" autofocus

    placeholder="Last name" /><br />

  <input type="submit" value="Submit" />

</form>
```

6. **Filename: multiple.html**
   Targeted toward selecting multiple files at once, rather than just a single file -- multiple is available in conjunction with other input types by separating individual values with a comma.

```
<form action="#" method="post">

 <label>Select images:</label>

   <input type="file" name="images" multiple />

 <input type="submit" />

</form>
```

End of Activity

Read Additional Resources

1. Week6-7_HTML5 Web Forms.pdf

Watch Video Resources

1. Week6-7_HTML Forms and JavaScript.mp4
2. Week6-7_JavaScript Form Validation.mp4

Internet References

1. https://w3schools.com
2. https://www.youtube.com/watch?v=ikR9DsGMUMc
3. https://www.youtube.com/watch?v=In0nB0ABaUk
4. https://visualstudiomagazine.com/articles/2012/01/01/web-forms-with-html5.aspx

CP1121