Contents

UNIT 2 – Forms in XHTML

Structure of Forms in XHTML

2.1 Learning Outcomes:

After the successful completion of this unit, the student will be able to:

- Explain the concept of form handling in XHTML.
- Explain the role of different form elements in XHTML in developing well-formed and standard web applications
- Create forms to collect different types of information from the users.

2.2 Forms in XHTML

Web pages are made interactive by the use of forms. Any online application collects data/information depending on the type of application like shopping, booking of bus tickets, railway reservations, ordering of food items, and many more by the use of forms. Data collected from the form is called form data and is sent/submitted to the web server for further processing.

There are various form elements like text fields, textarea, radio buttons, checkboxes, drop-down menus, etc that are used for collecting information from the users. These form elements are called form controls or widgets. The controls may collect single-line or multi-line text information. Every form requires a submit button. On form submission, form data is encoded and sent to the server for processing.

2.2.1 Form tag

In XHTML, <form> element is used to create a form. All controls or widgets used in the form are embedded in the opening <form> tag and closing </form> tag. It acts as a container that holds all controls within it.

For example, the syntax for creating a form in XHTML is as follows-

```
<form  form-attributes >
....  Form elements
</form>
```

2.2.2 Form Attributes

In XHTML, the <form> element contains several attributes that control the submission of form data. The most commonly used attributes with the form are -

- name/id
- method
- action
- target
- enctype

1. name/id –

   The name/id for the form can be given by using this attribute, which can later be used to reference or access the form in JavaScript or Cascading Style sheets. It is used to uniquely identify the form.

2. method –

   It is used to specify the HTTP method used when submitting the form data to the server. The methods used can be either GET or POST. By default GET method is used to send data to the server when the method attribute is not used. However, the POST method has advantages over the GET method.

   Characteristics of the GET method :

   - The form data is appended to the URL, in the form of name/value pair.
   - The GET method is never used when sending sensitive data to the server as it is visible in the URL
   - There is a limitation in sending data using GET as the length of a URL is limited to 2048 characters
   - For non-secure data GET method can be used.

Characteristics of the POST method:

- The form data is appended inside the body of the HTTP request
- For sending sensitive data to the server POST method is used as the submitted form data is not shown/visible in the URL.
- There is no size limitation in sending data using POST and can be used to send large amounts of data.
- For secure data POST method is used.

3. action –

   It is used to specify the path or URL of the document that processes or collects the form data when the user clicks the submit button.

4. target –

   It is used to specify the destination where the form submission response should be displayed. The target attribute can take the values

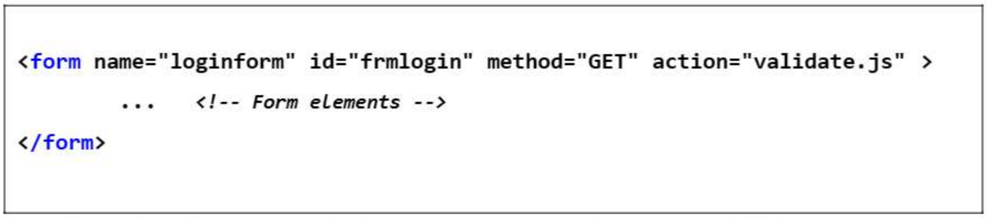
   - "_blank" - opens in a new window/tab
   - "_self" - opens in the same window.

5. enctype –

   It is used to specify the format of encoding form data before submitting it to the server. The possible values are

   - "application/x-www-form-urlencoded" - This is the default encoding type, all characters are encoded before data is sent to the server. Here spaces are converted to "+" and special characters are converted to ASCII Hex values.
   - "multipart/form-data" - This value is used when a file is to be uploaded to the server through a form.
   - "text/plain" – This is used when data encoding is not required and is rarely used.

Example 1: A form declaration with name, id, method, and action attributes in XHTML



```
<form name="loginform" id="frmlogin" method="GET" action="validate.js" >
        ...    <!-- Form elements -->
</form>
```

Figure 2.1: Declaration of <form> element with attributes in an XHTML document

The file **"validate.js"** in the action attribute is supposed to be executed on submission of the form.

Example 2: A form declaration with name, method, action, and enctype attributes in XHTML –

```
<form name="upldform" method="POST" enctype="multipart/form-data"
                                     action="upload.php" >
            ...
            ...   <!-- Form elements -->
</form>
```

Figure 2.2: Declaration of <form> element with enctype attribute in an XHTML document

The **"upload.php"** file in the action attribute is supposed to be executed on submission of the form. The **enctype="multipart/form-data"** is used when a file is to be uploaded on the server.

2.3 Form Elements/Controls

In XHTML, <form> element acts as a container for several other controls or widgets that are used for taking inputs from the user. The most commonly used controls are – <label>, <input>, <textarea>, <select>, <button>, etc

Table 2.1: Commonly used controls/widgets within <form> element in XHTML

Name of the control/widget	Description
<input>	It is used to take input from the user and is the most fundamental control used in forms
<label>	It is used to define a label for other controls in the form
<textarea>	It is used to input multiple lines of text
<button>	It is used to create a button that can be used for handling events
<select>	It is used to create a drop-down menu

2.3.1 Input element <input>

In the XHTML form, most of the controls are created using the <input> element. It is used for text fields, radio buttons, checkboxes, passwords, clickable buttons, and action buttons like submit and reset. The commonly used attributes with <input> element are :

Table 2.2: Commonly used attributes with <input> element in XHTML

Attribute names	Description
name	It is used to give a name to the control. This name can be referenced or accessed in JavaScript or PHP script during processing.
id	It is used to give an "**id**" to the control, that can be uniquely identified. The "**id**" can be referenced or accessed by other controls in the form and also in JavaScript or PHP script during processing.
type	It can take the values "text", "password", "radio", "checkbox", "button", "submit", "reset", etc. The default value for the "type" attribute is "text". Depending on the type of control to be used in the form, the value of "type" is set.
value	It is used to provide an initial or default value to the input text fields like textbox, radio buttons, and checkboxes but in the case of action buttons, the value assigned is displayed on the buttons itself.
size	It is used to specify the size/width of the text box to be displayed in terms of characters. The default size is 20 characters. If the user enters more characters than the size, it gets scrolled. This attribute is used in case of <input type="**text**"> and <input type="**password**">
maxlength	It is used to specify the maximum number of characters that the text field can accept from the user. This attribute is used in case of <input type="**text**"> and <input type="**password**">
placeholder	It is used to specify text within the input field to inform the user about the expected input. This attribute is used in case of <input type="**text**"> and <input type="**password**">
required	It is a boolean attribute which indicates the user must enter input in that field
disabled	It is used when a default value has to be set and does not require any interaction with the user.

The type of control to be defined using the <input> element depends upon the value set for its "type" attribute.

Table 2.3: Commonly used values for "type" attribute of <input> element in XHTML

Values used in the "type" attribute	Description
`<input type="text">`	Defines a single-line of text field called a textbox
`<input type="password">`	Defines a single-line text field, the input/password entered is not visible to the user
`<input type="radio">`	Defines a radio button. (Only one radio button can be selected from a group of radio buttons at a time)
`<input type="checkbox">`	Defines a checkbox. (Many checkboxes can be selected/ checked at a time)
`<input type="button">`	Defines a clickable button used for handling events
`<input type="submit">`	Defines a submit button used for form submission
`<input type="reset">`	Defines a reset button used for resetting the controls in the form

- <input type="text">

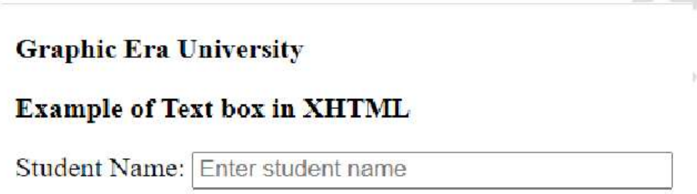
It is used to define textboxes that take a single line of input. For example to create search boxes or to enter information where a single line of input is required.

Example: Create a textbox to enter the student’s name. The length of the student’s name can be a maximum of 30 characters and it is a mandatory field.
For creating a textbox, `<input type="text">` is used as follows

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of textbox in XHTML </title>
</head>
```

```
<body>
     <p><b> Graphic Era University </b></p>
     <p><b> Example of Text box in XHTML </b></p>

      <form method="GET">
        Student Name: <input type="text" name="txtstudname" size="35"
                       maxlength="30"  placeholder="Enter student name"
                       required />
       </form>
</body>
</html>
```

Output:



Graphic Era University

Example of Text box in XHTML

Student Name: Enter student name

Figure 2.3: Use of <input type="text"> element in a <form> in XHTML

- <input type="password">

It is also used to define textboxes that take a single line of input but here the characters are masked as the user enters them.

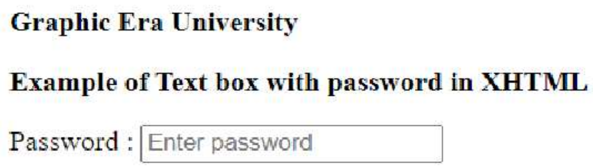
Example: Create a textbox to enter a password. The password should be a maximum 8 characters and it is a mandatory field.

For creating a textbox to enter password, `<input type="password">` is used as follows

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of password in XHTML </title>
</head>
<body>
    <p><b> Graphic Era University </b></p>
    <p><b> Example of Text box with password in XHTML </b> </p>
```

```
      <form method="GET">
         Password : <input type="password" name="txtpassword" size="20"
                               maxlength="8"  placeholder="Enter password"
                               required />
     </form>
  </body>
  </html>
```

Output:



Graphic Era University

Example of Text box with password in XHTML

Password : Enter password

Figure 2.4: Use of `<input type="password">` element in a <form> in XHTML

- <input type="radio">

It is used to create radio buttons. They are used when only one out of many options is to be selected. It allows the user to select only one option. The value of the attribute "name" should be the same for all the radio buttons within the group to make them mutually exclusive. The attribute "Checked" is used with a radio button to select it by default.

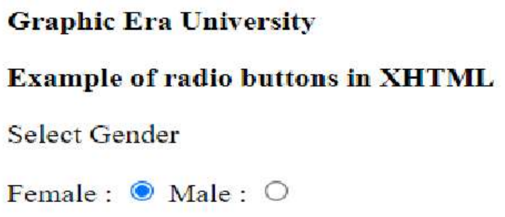
Example: Create radio buttons to select gender.

For radio buttons, `<input type="radio">` is used as follows

```
  <?xml version="1.0" encoding="utf-8" ?>
  <!DOCTYPE html>
  <html>
  <head>
      <title>Example of radio buttons in XHTML </title>
  </head>
  <body>
      <p><b> Graphic Era University </b></p>
     <p><b> Example of radio buttons in XHTML </b> </p>

     <form method="GET">
       <p> Select Gender </p>
       Female:<input type="radio" name="txtstudgender" value="F" checked />
```

```
        Male  :<input type="radio" name="txtstudgender" value="M" />
      </form>
  </body>
  </html>
```

Output:



Graphic Era University

Example of radio buttons in XHTML

Select Gender

Female : ● Male : ○

Figure 2.5: Use of `<input type="radio">` element in a <form> in XHTML

- <input type="checkbox">

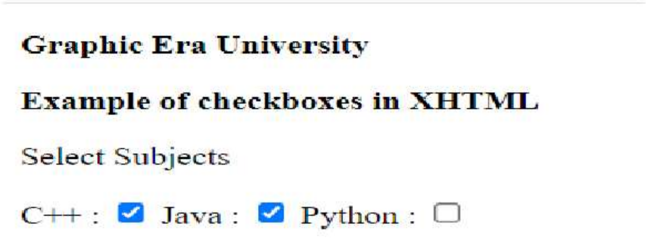
It is used to create checkboxes. They are used when more than one option is to be selected. The attribute "Checked" is used with a checkbox to select it by default.

Example: Create checkboxes to select subjects offered.

For checkboxes, `<input type="checkbox">` is used as follows

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of check boxes in XHTML </title>
</head>
<body>
 <p><b> Graphic Era University </b></p>
 <p><b> Example of checkboxes in XHTML </b> </p>
<form>
   <p> Select Subjects </p>
    C++ : <input type="checkbox" name="chkcplus" value="cplus" checked />
   Java : <input type="checkbox" name="chkjava" value="java" />
  Python: <input type="checkbox" name="chkpython" value="python" />
</form>
</body>
</html>
```

Output:



Graphic Era University

Example of checkboxes in XHTML

Select Subjects

C++ : ☑ Java : ☑ Python : □

Figure 2.6: Use of `<input type="checkbox">` element in a <form> in XHTML

- <input type="button">

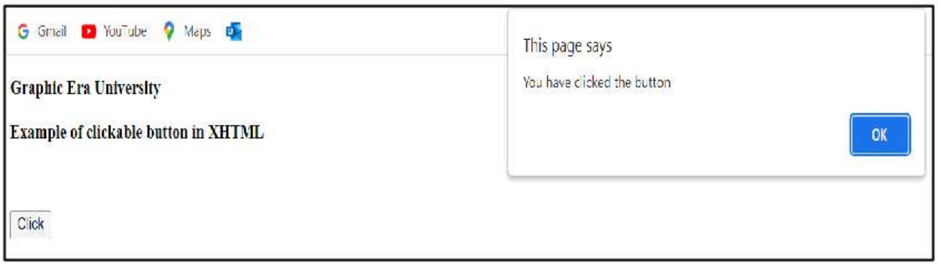
This is used to create a clickable button or a simple push button that can be used to handle events. It triggers a script that handles the event. The script usually is JavaScript.

Example: Create a button, which on clicked displays a message.

For buttons, `<input type="button">` is used as follows

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of buttons in XHTML </title>
</head>
<body>
  <p><b> Graphic Era University </b></p>
   <p><b> Example of clickable button in XHTML </b> </p><br/>
   <br/>
   <form>
     <!-- Value "click" is displayed on the button, on clicking the
          button click event is handled by displaying a message
          using JavaScript code
      -->
      <input type="button" value="Click"
             onclick="alert('You have clicked the button')" />
   </form>
</body>
</html>
```

Output:



Graphic Era University

Example of clickable button in XHTML

This page says

You have clicked the button

Figure 2.7: Use of `<input type="button">` element in a <form> in XHTML

- <input type="submit"> and <input type="reset">

Action buttons like submit and reset are created using the <input> element.

The Reset button is used to reset all the controls in the form to their initial states.

The Submit button is used for submitting form data to the form handler given in the action attribute in the <form> element. The form-handler is usually a JavaScript or a PHP script on the server, that processes the form data. The form data is encoded before it is sent to the server. The server on the other hand executes the server-resident script and returns the response to the user.

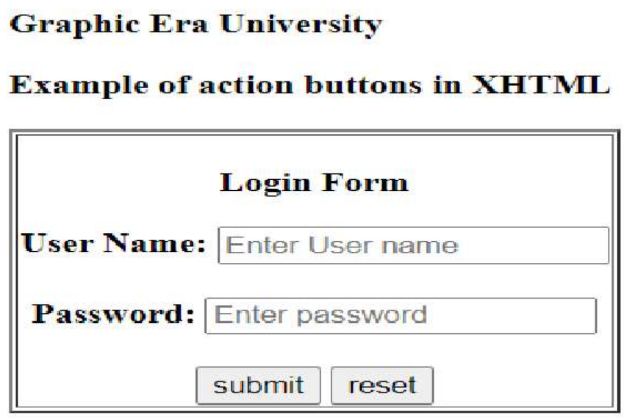
Example: To create a user interface for a login form

For submit and reset buttons, `<input type="submit">` and `<input type="reset">` is used as follows -

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of action buttons in XHTML </title>
</head>

<body>
   <p><b> Graphic Era University </b></p>
   <p><b> Example of action buttons in XHTML </b> </p>

<form action="validate.js">
   <table border="2">
```

```
    <tr>
      <th>
        <p><b> Login Form </b> </p>
        <label>  User Name: </label>
        <input type="text" id="txtusrname" maxlength="25"
                        placeholder="Enter User name" />
         <br />
         <br />
        <label>  Password: </label>
        <input type="password" id="txtpasswd" maxlength="8"
                      placeholder="Enter password" /> <br/><br/>

 <!-- value "submit" is displayed on the button -->
         <input type="submit" value="submit" />

<!-- value "reset" is displayed on the button -->
         <input type="reset" value="reset" />
     </th>
   </tr>
</table>
</form>

</body>
</html>
```

Output:



Graphic Era University

Example of action buttons in XHTML

Login Form

User Name: Enter User name

Password: Enter password

Figure 2.8: Use of `<input type="submit">` **and** `<input type="reset">` element in XHTML

2.3.2 Labels <label>

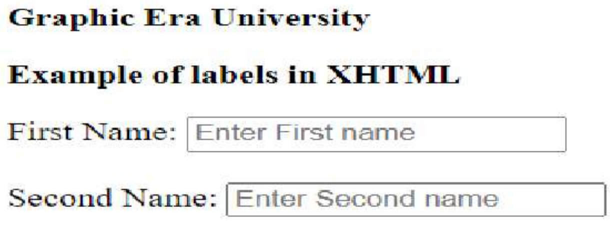
Label control <label> is used to define a label for other controls in the form.

Example : `<label>` Student Name `</label>`

For creating labels, `<label>` is used as follows

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of labels in XHTML </title>
</head>
<body>
    <p><b> Graphic Era University </b></p>

    <p><b> Example of labels in XHTML </b> </p>

    <form>
     <label> First Name: </label>
     <input type="text" id="txtstdfname"  maxlength="25"
                       placeholder="Enter First name" />
     <br />
     <br />
     <label> Second Name: </label>
     <input type="text" id="txtstdsname" maxlength="25"
                       placeholder="Enter Second name" />

    </form>
</body>
</html>
```

Output:



Graphic Era University

Example of labels in XHTML

First Name: Enter First name

Second Name: Enter Second name

Figure 2.9: Use of `<label>` element in a <form> in XHTML

2.3.3 Textarea <textarea>

Textarea control <textarea> is used when more than one line of text/information is required to be entered by the user. It is used for multiple-line inputs. The attributes used with <textarea> are different from those used with single-line text fields.

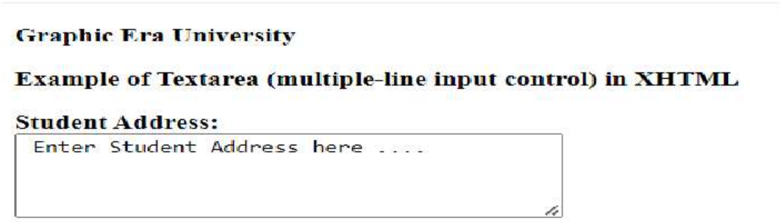
Table 2.4: Commonly used attributes with <textarea> element in XHTML

Attribute name	Description
name	Defines a name for the Textarea control
rows	It is used to specify the number of rows of the Textarea box
cols	It is used to specify the number of columns of the Textarea box

Example: Create a form with controls to enter the address of the student.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of textarea in XHTML </title>
</head>
<body>
    <p><b> Graphic Era University </b></p>
    <p><b> Example of Textarea in XHTML </b> </p>
    <form >
       <b> Student Address: </b> <br/>
       <textarea name="txtstudaddr" rows="4"  cols="40" >
            Enter Student Address here ....
       </textarea>
    </form>
</body>
</html>
```

Output:



Graphic Era University

Example of Textarea (multiple-line input control) in XHTML

Student Address:

```
Enter Student Address here ....
```

Figure 2.10: Use of **<textarea>** element in a <form> in XHTML

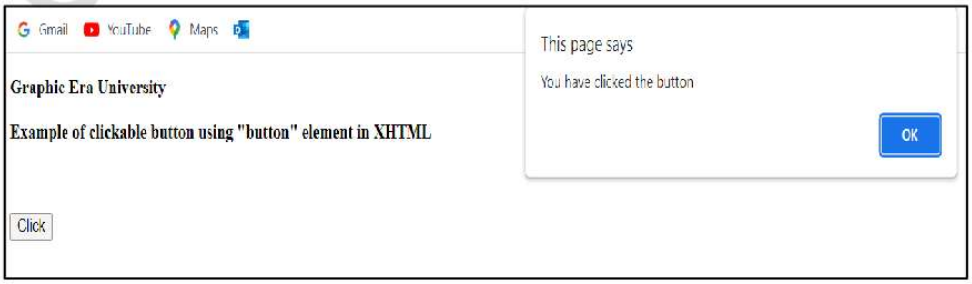
2.3.4 Button

The <button> element is used to define a clickable button and handle events.

Example: To create a button that displays a message on click.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of buttons in XHTML </title>
</head>
<body>
    <p><b> Graphic Era University </b></p>

     <p><b> Example of clickable button using "button" element in XHTML </b>
      </p><br/>
       <br/>

     <form>
     <!-- Value "click" is displayed on the button, on clicking the
          button click event is handled by displaying a message
          using JavaScript code
       -->
     <button type="button" onclick="alert('You have clicked the button')" >
         Click
     </button>

     </form>
</body>
</html>
```

Output :



Graphic Era University

Example of clickable button using "button" element in XHTML

This page says

You have clicked the button

Figure 2.11: Use of **<button>** element in a <form> in XHTML

2.3.5 Drop-down menu - <select>

A drop-down menu can be created by using <select> element in XHTML. Radio buttons and checkboxes are usually used when collecting multiple-choice data. Radio buttons can be used when only one option has to be opted from many and checkboxes are used when many multiple options are to be selected. But when the number of possible options is large then the display of checkboxes on the screen also increases and is not a better choice. In such situations, a drop-down menu is used and in XHTML it is created by using <select> element.

Two types of drop-down menus that can be created -

1. Similar to radio buttons – where only one option/item can be selected at a time from a group of options/items.
2. Similar to checkboxes - where one or many options/items can be selected at a time from a group of options/items.

By default, a drop-down menu has the features of a radio button where only one item can be selected, but by using the attribute "multiple" with <select>, it can be made to behave like checkboxes where multiple options can be selected.

Syntax is

```
<select attributes>
      <option attributes > item1 </option>
      <option attributes > item2 </option>
      . . .
</select>
```

The most commonly used attributes with <select> element are –

Table 2.5: The most commonly used attributes with <select> element

Attribute name	Description
name	Defines a name for the <select> element
size	It is used to specify the number of items from the drop-down menu that are to be made visible (displayed) to the user on the screen. A scrolling list box is displayed.
multiple	It allows the user to select more than one item at a time.

The most commonly used attributes with <option> elements are

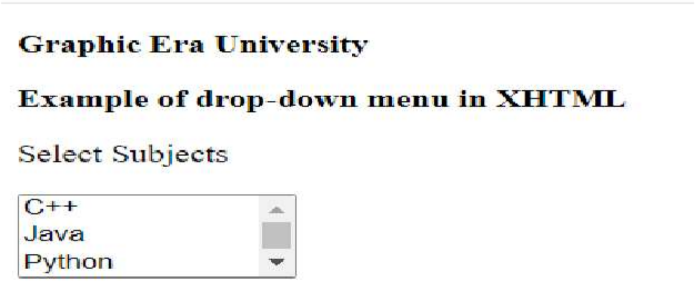
Table 2.6: The most commonly used attributes with <option> element

Attribute name	Description
value	It is the value that is to be used when the present option is selected
selected	It is used to define a pre-selected value.

Example: Use of <select> in XHTML

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html>
<head>
    <title>Example of drop-down menu in XHTML </title>
</head>
<body>
    <p><b> Graphic Era University </b></p>
    <p><b> Example of drop-down menu in XHTML </b> </p>
   <form>
   <!-- The drop-down menu here has 5 subjects but as
        the "size" attribute value is set to 3 only 3 subjects are visible
        in the drop-down.
        The attribute "multiple" allows to select more than one subject
       by holding down the shift key
      The attribute "selected" allows that option to be pre-selected
   -->
      <p> Select Subjects </p>
        <select name="subjectlist" size="3" multiple>
             <option  value="Cplus" > C++ </option>
             <option  value="Java" > Java  </option>
             <option  value="Python" > Python </option>
             <option  value="DS" > Data Structures </option>
             <option  value="OS" selected > Operating System </option>
        </select>
    </form>
</body>
</html>
```

Output :



Graphic Era University

Example of drop-down menu in XHTML

Select Subjects

Figure 2.12: Use of **<select>** element for creating a drop-down in an <form> in XHTML

2.4 Creating a Form in XHTML – Simple Example

Example: Creating a form with types of form elements – Textbox, password, radio buttons, checkboxes, drop-down, submit and reset buttons, clickable buttons using <input> and <button> tags are shown as follows -

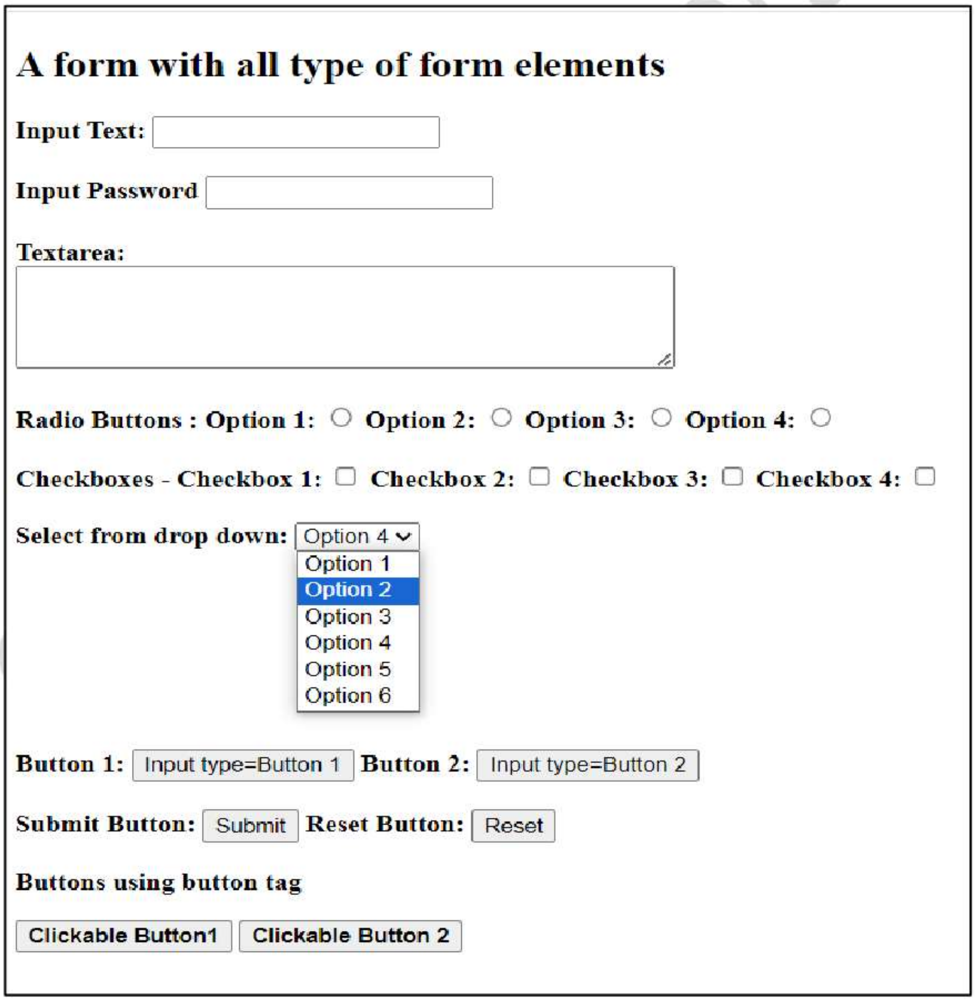
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html >
<html>
<head>
  <title>Form Controls Example</title>
</head>
<body>
  <h2> A form with all type of form elements   </h2>

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

<!-- Input - textbox -->
    <label for="input_text"><b>Input Text: </b></label>
    <input type="text" id="input_text" name="input_text" /><br /><br />

<!-- Input -  password  -->
   <label for="input_password"><b>Input Password </b></b></label>
   <input type="password" id="input_password" name="input_password" />
   <br /><br />
```

```
<!-- Input - textarea - -->
<label for="textarea_input"><b>Textarea: </b></b></label><br />
<textarea  id="textarea_input"  name="textarea_input"  rows="4"  cols="50">
</textarea>
<br /><br />

<!-- Input - radio button -->
 <label for="input_radiooption1"><b> Radio Buttons : Option 1:</b></label>
 <input type="radio" id="input_radiooption1" name="input_radiooption"
        value="option1" />

 <label for="input_radiooption2"><b> Option 2:</b></label>
 <input type="radio" id="input_radiooption2" name="input_radiooption"
        value="option2" />

 <label for="input_radiooption3"><b> Option 3:</b></label>
 <input type="radio" id="input_radiooption3" name="input_radiooption"
        value="option3" />

 <label for="input_radiooption4"><b> Option 4:</b></label>
 <input type="radio" id="input_radiooption4" name="input_radiooption"
        value="option4" />
 <br /><br />

<!-- Input - checkboxes  -->
  <label for="input_checkbox1"><b>Checkboxes  - Checkbox 1: </b></label>
  <input type="checkbox" id="input_checkbox1" name="input_checkbox1" />

  <label for="input_checkbox2"><b>Checkbox 2: </b></label>
  <input type="checkbox" id="input_checkbox2" name="input_checkbox2" />

  <label for="input_checkbox3"><b>Checkbox 3: </b></label>
  <input type="checkbox" id="input_checkbox3" name="input_checkbox3" />

  <label for="input_checkbox4"><b>Checkbox 4: </b></label>
  <input type="checkbox" id="input_checkbox4" name="input_checkbox4" />
  <br /><br />
```

```
<!-- Input - drop-down  -->

    <label for="input_select"><b>Select from drop down:</b></label>
    <select id="input_select" name="input_select">
      <option value="option1">Option 1</option>
      <option value="option2">Option 2</option>
      <option value="option3">Option 3</option>
      <option value="option4">Option 4</option>
      <option value="option5">Option 5</option>
      <option value="option6">Option 6</option>
    </select>
    <br /><br /><br /> <br /><br /> <br /><br /> <br />

<!-- Input - Button -->

   <label for="input_button1"><b> Button 1: </b></label>
   <input type="button" id="input_button1" name="input_button1"
          value="Input type=Button 1"
          onclick="alert('Clickable Button 1 clicked')" />


   <label for="input_button2"><b>Button 2: </b></label>
   <input type="button" id="input_button2" name="input_button2"
          value="Input type=Button 2"
          onclick="alert('Clickable Button 2 clicked')" />
   <br /><br />

<!-- submit and reset buttons -->

   <label for="submit_button"><b>Submit Button:</b></label>
   <input type="submit" id="submit_button" name="submit_button"
          value="Submit" />

<label for="reset_button"><b>Reset Button: </b></label>
<input type="reset" id="reset_button" name="reset_button" value="Reset" />
<br />
```

```
    <!-- buttons -->
      <p> <b>Buttons using button tag </b><p>
      <button type="button" onclick="alert('You have clicked the button 1')" >
          <b> Clickable Button1 </b>
      </button>
      <button type="button" onclick="alert('You have clicked the button 2')" >
          <b>  Clickable Button 2 </b>
      </button>

    </form>
   </body>
</html>
```

Output :



A form with all type of form elements

Input Text:

Input Password

Textarea:

Radio Buttons : Option 1: ○ **Option 2:** ○ **Option 3:** ○ **Option 4:** ○

Checkboxes - Checkbox 1: □ **Checkbox 2:** □ **Checkbox 3:** □ **Checkbox 4:** □

Select from drop down: Option 4

Button 1: Input type=Button 1 **Button 2:** Input type=Button 2

Submit Button: Submit **Reset Button:** Reset

Buttons using button tag

Figure 2.13: Use of all form elements in a XHTML form

2.5 Self-Assessment Questions

Q1. What is the purpose of using forms in XHTML? Which element is used to create forms in XHTML? Give an example. [5 marks, L2]

Q2. Explain the attributes used with forms in XHTML with a suitable example. [8 marks, L2]

Q3. Explain the characteristics of GET and POST methods used in XHTML.[3 marks, L2]

Q4. Which are the controls that are commonly used with form elements in XHTML. Explain in brief. [8 marks, L2]

Q5. List and explain in brief the commonly used attributes with <input> element in XHTML. [8 marks, L2]

Q6. List and explain the different types of controls that can be created using the "type" attribute of <input> element. [8 marks, L2]

Q7. What is the purpose of using the "label" element in an XHTML form? Give a suitable example and the necessary code. [3 marks, L2]

Q8. What is the difference between textbox and textarea elements used with forms in XHTML.? Give suitable examples [5 marks, L2]

Q9. What is the use of the "Submit" button in XHTML form? Explain with a suitable example and write the required code. [6 marks, L2]

Q10. Explain how to create a drop-down menu in XHTML. Explain with a suitable example and write the necessary code. [8 marks, L2]

Q11. What is the difference between radio buttons and checkboxes? Explain the use of radio buttons and checkboxes with suitable examples and the required code. [8 marks, L2]

Q12. Write short notes on the following w.r.t forms in XHTML [5 marks each, L2]

- Textboxes and password elements
- Textarea elements
- Radio buttons
- Checkboxes
- Drop-down menu
- Submit and Reset buttons
- GET and POST methods

Q13. Design a Registration form for registration of participants in inter-colligate sports competitions held by "Graphic Era University". [Use textbox, textarea, radio buttons, checkboxes, drop-down menu, submit and reset buttons, and the required controls to design the form]

Q14. Design a form for registration of the 'C' Programming Quiz competition conducted for the First Year B.Tech Students of Graphic Era University. [Use the required control in the form]

2.6 Self-Assessment Activities

A1. Design a form for registration of IN patients in a hospital. Use appropriate form elements to serve their purpose.

A2. Design an online order form for a fruit shop. [Use the required elements in the form. [For better presentation and alignment of data XHTML tables can be used]

2.7 Multiple-Choice Questions

1. In XHTML, a form is created by using ________ element. [1 mark, L1]
   a) <input type=”form” >
   b) <form>
   c) <label type=”form”>
   d) None of the above

2. A label is used in a form to ___________. [1 mark, L1]
   a) add a button
   b) change the color of the form
   c) Define a label for other controls in a form.
   d) None of the above

3. The <input type=__________ > in an XHTML form is used to mask the input entered by a user. [1 mark, L1]
   a) “text”
   b) “password”
   c) “placeholder”
   d) “textarea”

4. The difference between “Radio Buttons” and “Checkboxes” is XHTML is _____. [1 mark. L1]
   a) Radio buttons allow only one selection, checkboxes allow multiple.
   b) Radio buttons allow multiple selections, Checkboxes allow only one.
   c) Radio buttons do not have a drop-down, Checkboxes have a drop-down.
   d) Radio buttons can be used in forms, Checkboxes cannot be used

5. The action attribute is used inside the _____ element in XHTML. [1 mark, L1]
   a) <input>
   b) <button>

c) <label>

d) <form>

6. Which of the following is incorrect w.r.t to GET method in XHTML? [1 mark, L2]

   a) GET is the default method

   b) GET method has limitations for sending data

   c) GET is more secure than POST when sending data to the server

   d) The form data is appended to the URL, in the form of name/value pair.

7. The attribute “rows” and “cols” is used with ____ element in XHTML form.[1 mark, L1]

   a) <input>

   b) <textarea>

   c) <select>

   d) <button>

8. ____ element is used for creating a drop-down list in XHTML. [1 mark, L1]

   a) <select>

   b) <dropdown>

   c) <menu>

   d) <input>

9. To select more than one item from the drop-down in XHTML ___ is used.[1 mark, L1]

   a) <text multiple >

   b) <option multiple>

   c) <select multiple>

   d) <input multiple>

10. ___ is the default type of the “type” attribute of <input> element. [1 mark, L1]

    a) image

    b) file

    c) password

    d) text

2.8 Key Answers to Multiple-Choice Questions

1. In XHTML, a form is created by using <form> element. [b]
2. A label is used in a form to define a label for other controls in a form. [c]

3. The <input type=**"password"** > in an XHTML form is used to mask the input entered by the user. [b]
4. The difference between "Radio Buttons" and "Checkboxes" is XHTML is radio buttons allow only one selection, checkboxes allow multiple.[a]
5. The action attribute is used inside the <form> element in XHTML.[d]
6. GET is more secure than POST when sending data to the server is incorrect w.r.t GET method in XHTML.[c]
7. The attribute "rows" and "cols" is used with <textarea> element in XHTML form.[b]
8. <select> element is used for creating a drop-down list in XHTML.[a]
9. To select more than one item from the drop-down in XHTML <select multiple> is used.[c]
10. Text is the default type of the "type" attribute of <input> element. [d]

2.9 Summary

In XHTML, <form> element is used to create a form. All controls or widgets used in the form are embedded in the opening <form> tag and closing </form> tag. It acts as a container that holds all controls within it.

For example, the syntax for creating a form in XHTML is as follows-

```
<form  form-attributes >
….  Form elements
</form>
```

In XHTML, the <form> element contains several attributes that control the submission of form data. The most commonly used attributes with the form are - name/id, method, action, target, enctype.

In XHTML, <form> element acts as a container for several other controls or widgets that are used for taking inputs from the user. The most commonly used controls are –

<label>, <input>, <textarea>, <select>, <button>, etc

Action buttons like submit and reset are created using the <input> element.

The Reset button is used to reset all the controls in the form to their initial states.

The Submit button is used for submitting form data to the form handler given in the action attribute in the <form> element. The form-handler is usually a JavaScript or a PHP script on the server, that processes the form data. The form data is encoded before it is sent to the server. The server on the other hand executes the server-resident script and returns the response to the user.

Textarea control <textarea> is used when more than one line of text/information is required to be entered by the user. It is used for multiple-line inputs. The attributes used with

<textarea> are different from those used with single-line text fields. The attributes used with <textarea> are name, rows, and cols.

A drop-down menu can be created by using <select> element in XHTML. Radio buttons and checkboxes are usually used when collecting multiple-choice data. Radio buttons can be used when only one option has to be opted from many and checkboxes are used when many multiple options are to be selected. But when the number of possible options is large then the display of checkboxes on the screen also increases and is not a better choice. In such situations, a drop-down menu is used and in XHTML it is created by using <select> element.

Two types of drop-down menus that can be created -

- Similar to radio buttons – where only one option/item can be selected at a time from a group of options/items.
- Similar to checkboxes - where one or many options/items can be selected at a time from a group of options/items.

By default, a drop-down menu has the features of a radio button where only one item can be selected, but by using the attribute "multiple" with <select>, it can be made to behave like checkboxes where multiple options can be selected.

2.10 Keywords

- Forms
- Input controls
- Textarea
- Submit
- Reset
- Drop-down <select>
- Buttons
- Form Attributes
- Form elements

2.11 Recommended resources for further reading

a. Essential Reading

1. Sebesta, R. W. (2010). Programming the World Wide Web (6th ed.), Pearson education.
2. Subramanian, V. (2019). Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node (2nd Ed.), Apress.

b. Recommended Reading

1. DT Editorial Services. (2016). HTML 5: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP & jQuery: Black Book, Dreamtech Press.
2. Koroliova, E. W. I., (2018). MERN Quick Start Guide: Build Web applications with MongoDB, Express.js, React and Node, Packt.

--*--