KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24



KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**Batch:A1 Roll No.: 16010422013 Experiment No.: 2**

**Aim**: To design website forms to accept data from the user through the HTML 5.0 form elements.

**Resources needed:**HTML 5.0 editor

**Theory:**

**Basics of HTML Forms:**

HTML forms contain **form elements**. Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

**For Example:**

**<input type="text">** defines a one-line input field for **text input.**

**<input type="radio">** defines a **radio button**.

The other input elements are:

● Checkboxes

● Button

● Textarea

● Select

**The different attributes of forms are:**

**The Action Attribute**: The **action attribute** defines the action to be performed when the form is submitted.The common way to submit a form to a server, is by using a submit button.Normally, the form is submitted to a web page on a web server.

For example:

<form**action="action\_page.php**">

**The Method Attribute:The**method attribute **specifies the HTTP method (**GET **or** POST**) to be used when submitting the forms:**

For example:

<formaction="action\_page.php"**method="get"**> or <form action="action\_page.php" **method="post"**>

**A history of HTML5 forms:**

The forms section of HTML5 was originally a specification titled Web Forms 2.0 that added new types of controls for forms. Started by Opera and edited by then-Opera employee Ian Hickson, it was submitted to the W3C in early 2005. The work was initially carried out under the W3C. It was then combined with the Web Applications 1.0 specification to create the basis of the breakaway Web Hypertext Application Technology Working Group (WHATWG) HTML5 specification.

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**Using HTML5 design principles**

One of the best things about HTML5 forms is that you can use almost all of these new input types and attributes right now. They don’t even need any shivs, hacks, or workarounds. That isn’t to say they’re all “supported” right now, but they do cool things in modern browsers that do support them-and degrade gracefully in browsers that don’t understand them. This is thanks to HTML5’s design principles. In this instance we’re specifically referring to the principle of graceful degradation. In essence, this means that there’s no excuse for not using these features right now. In fact, it means you’re ahead of the curve.

**HTML5 form attributes**

There are 14 new attributes provided by HTML5

placeholder autofocus

autocomplete required

pattern list

multiple novalidate

formnovalidate form

formaction formenctype

formmethod formtarget

**1. placeholder**

First up is the placeholder attribute, which allows us to set placeholder text as we would currently do in HTML4 with the value attribute. It should only be used for short descriptions. For anything longer, use the title attribute. The difference from HTML4 is that the text is only displayed when the field is empty and hasn’t received focus. Once the field receives focus (e.g., you click or tab to the field), and you begin to type, the text simply disappears. It’s very similar to the search box you see in Safari (see Figure 1).



Figure 1. Browser search box in Safari without and with focus

Let’s have a look at how to implement the placeholder attribute.

**<input type="text"name="user-name"id="user-name"placeholder="at least 3 characters">**

Figure 2 shows the placeholder attribute working in Chrome.



Figure 2. Placeholder attribute support in Chrome, unfocused and focused

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**2. autofocus**

autofocus does exactly what it says on the tin. Adding it to an input automatically focuses that field when the page is rendered. It is a Boolean attribute (except if you are writing XHTML5; see the note) and is implemented as follows:

**<input type="text"name="first-name"id="first-name" autofocus>**

**3. autocomplete**

The autocomplete attribute helps users complete forms based on earlier input. The default state is set to on. This means that generally we won’t have to use it. However, if you want to insist that a form field be entered each time a form is completed (as opposed to the browser autofilling the field), you would implement it like so:

**<input type="text"name="tracking-code"id="tracking-code"autocomplete="off">**

The autocomplete state on a field overrides any autocomplete state set on the containing form element.

**4. required**

The required attribute doesn’t need much introduction; likeautofocus, it does exactly what you’d expect. By adding it to a form field, the browser requires the user to enter data into that field before submitting the form. required is a Boolean attribute, like autofocus. Let’s see it in action.

**<input type="text"id="given-name"name="given-name" required>**

**New Input Types in HTML5**

● color

● date

● datetime

● datetime-local

● email

● month

● number

● range

● search

● tel

● time

● url

● week

**The new Elements added by HTML5**

**list and the datalist element**

The list attribute enables the user to associate a list of options with a particular field. The value of the list attribute must be the same as the ID of a datalist element that resides in the

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

same document. The following example shows how list and datalist are combined (see Figure )

**<label>Your favorite fruit:**

**<datalist id="fruits">**

**<option value="Blackberry">Blackberry</option>**

**<option value="Blackcurrant">Blackcurrant</option>**

**<option value="Blueberry">Blueberry</option>**

**<!-- … -->**

**</datalist>**

**If other, please specify:**

**<input type="text" name="fruit" list="fruits">**

**</label>**

By adding a select element inside the datalist you can provide superior graceful degradation than by simply using an option element.

**<label>Your favorite fruit:**

**<datalist id="fruits">**

**<select name="fruits">**

**<option value="Blackberry">Blackberry</option>**

**<option value="Blackcurrant">Blackcurrant</option>**

**<option value="Blueberry">Blueberry</option>**

**<!-- … -->**

**</select>**

**If other, please specify:**

**</datalist>**

**<input type="text" name="fruit" list="fruits">**

**</label>**

Browser support for list and datalist is currently limited to Opera 9.5+ (see Figure 5), Chrome 20+, Internet Explorer 10 and Firefox 4+.



Figure 3 :Thedatalist element rendered in Opera

**Attributes of the Form tag:**

● Formaction

● Formenctype

● Formmethod

● Formtarget

● Novalidate

● formnovalidate

The novalidate and formnovalidate attributes indicate that the form shouldn’t be validated when submitted. They are both Boolean attributes. formnovalidate can be applied to submit or image input types. The novalidate attribute can be set only on the form element. The following example shows how to use formnovalidate:

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**<form action="process.php">**

**<label for="email">Email:</label>**

**<input type="text" name="email"value="gordo@example.com">**

**<input type="submit" formnovalidate value="Submit">**

**</form>**

And this example shows how to use novalidate:

**<form action="process.php" novalidate>**

**<label for="email">Email:</label>**

**<input type="text" name="email"value="gordo@example.com">**

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

**</form>**

**Activity:**

Design a form (eg. Registration form/feedback form/admission form etc) with HTML 5.0 new form features.

**Results: (Program printout with output / Document printout as per the format)** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Employment Application Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

form {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 80%;

max-width: 600px;

}

label {

display: block;

margin-bottom: 8px;

font-weight: bold;

}

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

input, select, textarea {

width: 100%;

padding: 8px;

margin-bottom: 16px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

textarea {

height: 100px;

}

button {

background-color: #4caf50;

color: #fff;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

</style>

</head>

<body>

<form action="process\_application.php" method="post" novalidate> <label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" placeholder="Your full name" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Your email" required>

<label for="phone">Phone:</label>

<input type="tel" id="phone" name="phone" placeholder="Your phone number" required>

<label for="position">Position Applied For:</label>

<input type="text" id="position" name="position" placeholder="Job position applied for" required>

<label for="resume">Resume/CV (PDF only):</label>

<input type="file" id="resume" name="resume" accept=".pdf" required>

<label for="coverLetter">Cover Letter:</label>

<textarea id="coverLetter" name="coverLetter" placeholder="Write your cover letter here" required></textarea>

<label for="availability">Availability:</label>

<select id="availability" name="availability" required>

<option value="fullTime">Full-Time</option>

<option value="partTime">Part-Time</option>

<option value="contract">Contract</option>

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

</select>

<label for="howDidYouHear">How did you hear about us?</label>

<input type="text" id="howDidYouHear" name="howDidYouHear" placeholder="E.g., Referral, Website, Social Media" required>

<label for="termsAndConditions">

<input type="checkbox" id="termsAndConditions" name="termsAndConditions" required>

I have read and agree to the Terms and Conditions

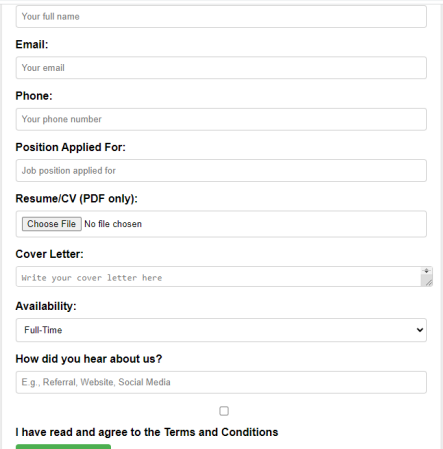
</label>

<button type="submit">Submit Application</button>

</form>

</body>

</html>



KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**Questions:**

**1. What is the use of multiple in list and datalist element?**

**The `multiple` attribute in the `<select>` element and the `datalist` element in HTML serves to enable users to select multiple options from a list. When used in a `<select>` element, it allows users to choose more than one option by holding down the Ctrl key (or Command key on Mac) while clicking on options. In the case of the `datalist` element, when combined with the `<input>` element, it provides a dropdown list of options, and users can select multiple values. This feature is particularly useful in scenarios where users need to choose from a set of options with multiple selections allowed, such as selecting multiple categories or preferences in a form. The `multiple` attribute enhances the flexibility and usability of web forms by accommodating scenarios that involve selecting multiple choices simultaneously.**

**2. What is the importance of pattern attribute?**

**The `pattern` attribute in HTML is crucial for enforcing a specific pattern or format for user input in form fields. This attribute is commonly used with text based input types, such as text and password fields, to ensure that the entered data adheres to a predefined pattern. For instance, it is often employed to validate email addresses, phone numbers, or custom formats like postal codes. By specifying a regular expression pattern, developers can define the expected structure of the input, enhancing data integrity and reducing errors.**

**The importance of the `pattern` attribute extends to user experience, as it provides immediate feedback to users when their input doesn't match the required format. This proactive validation contributes to a more user-friendly interface, guiding users in providing accurate information. Moreover, it helps prevent invalid data submissions, reducing the likelihood of processing errors on the server side. In summary, the `pattern` attribute is crucial for improving data accuracy, user experience, and overall form reliability by enforcing specific input patterns.**

**3. What are the three types of button that can be used in form?**

**In HTML forms, there are three main types of buttons: submit buttons, reset buttons, and regular buttons. Submit buttons, created with the `<input>` element and the `type="submit"` attribute, are crucial for triggering form submission, allowing user-entered data to be processed on the server. Reset buttons, generated with the `<input>` element and the `type="reset"` attribute, provide users with the option to clear or reset form fields to their default values. However, they are less common in modern forms due to potential data loss concerns. Regular buttons, formed with the `<button>` element, offer versatility for custom actions using JavaScript. Unlike submit and reset buttons, regular buttons lack predefined behaviors, making them suitable for various custom functionalities within a form.**

**Outcomes: Create webpages using html5 and css.**

KJSCE/IT/SY-BTech/SEM IV/WP1/2023-24

**Conclusion:**

**In conclusion, this experiment aimed to design website forms to accept user data through HTML 5.0 form elements. The basics of HTML forms, including various input types and attributes, were explored. The experiment covered the history of HTML5 forms and emphasized the importance of HTML5 design principles, such as graceful degradation. Additionally, new HTML5 form attributes and elements, such as placeholder, autofocus, autocomplete, and datalist, were implemented to enhance form functionality. The outcomes included the successful creation of an employment application form, incorporating HTML5 features for improved user experience and data validation. (Conclusion to be based on the outcomes achieved)**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**

**References:**

**Books/ Journals/ Websites:**

● "HTML5: Black Book", Dreamtech Publication.

● "Web Technologies: Black Book", Dreamtech Publication.

● http://www.w3schools.com