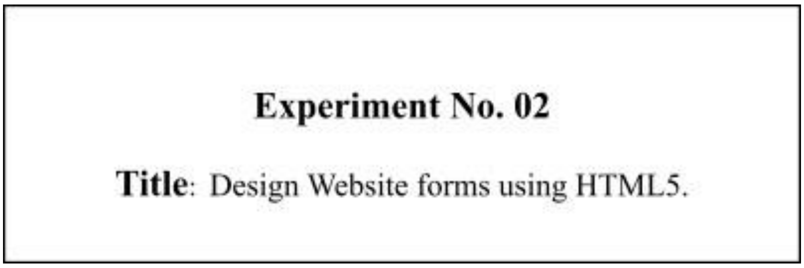


Experiment No. 02

Title: Design Website forms using HTML5.

Batch:B1 Roll No.:16010420133 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:Themethod 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.

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).

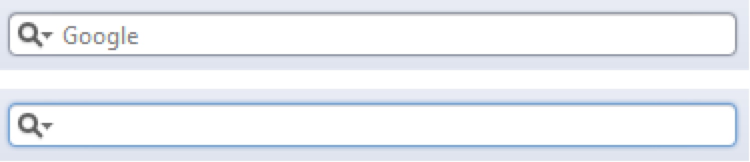


Figure 1. Browser search box in Safari without and with focus
Let's have a look at how to implement the placeholder attribute.

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

Figure 2 shows the placeholder attribute working in Chrome.

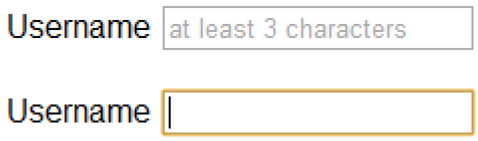


Figure 2. Placeholder attribute support in Chrome, unfocused and focused

2. <u>autofocus</u>

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. <u>autocomplete</u>

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. <u>required</u>

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

<u>The new Elements added by HTML5</u>

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

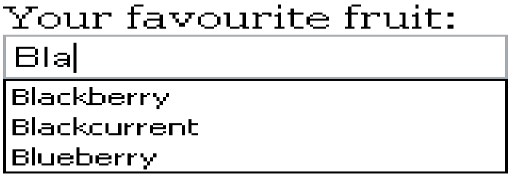
same document. The following example shows how list and datalist are combined (see Figure)

```
<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+.



Your favourite fruit:

Bla

Blackberry
Blackcurrent
Blueberry

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:

```
<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>
```

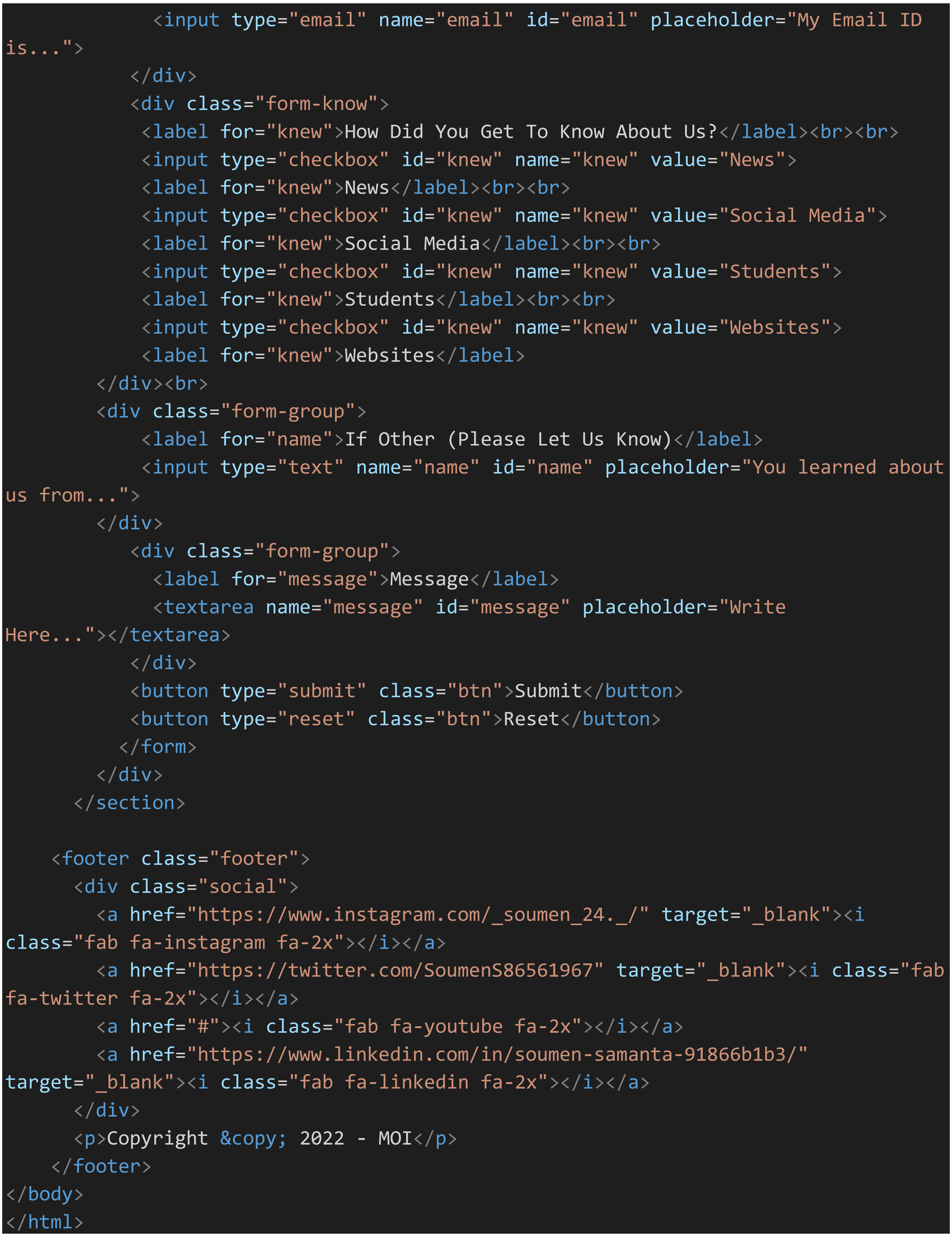
Activity:

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)

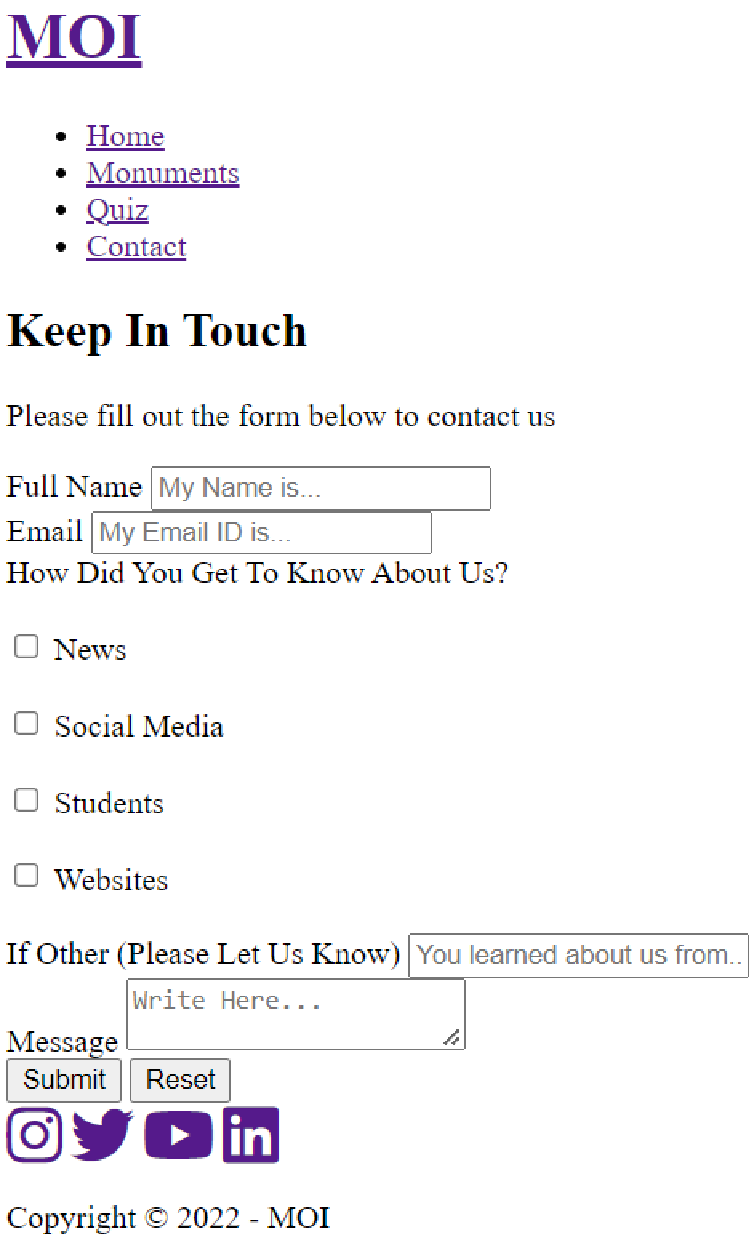
```
<!DOCTYPE html>
<html lang="en">
<head>

<title>MOI | Contact</title>
</head>
<body>
    <header>
      <div class="navbar top">
              <h1 class="logo"><a href="index.html">MOI</a></h1>
        <nav>
              <ul>
                <li><a href="index.html">Home</a></li>
                <li><a href="monument.html">Monuments</a></li>
                <li><a href="quiz.html">Quiz</a></li>
                <li><a class="current" href="contact.html">Contact</a></li>
              </ul>
        </nav>
      </div>
      <section id="contact-form" class="py-3">
        <div class="container">
          <h1 class="l-heading"><span class="text-primary">Keep In
Touch</span></h1>
          <p class="l-para">Please fill out the form below to contact us</p>
          <div class="underline">
          </div>
          <form action="process.php">
            <div class="form-group">
              <label for="name">Full Name</label>
              <input type="text" name="name" id="name" placeholder="My Name is...">
            </div>
            <div class="form-group">
             <label for="email">Email</label>
```



```
             <input type="email" name="email" id="email" placeholder="My Email ID
is...">
           </div>
           <div class="form-know">
            <label for="knew">How Did You Get To Know About Us?</label><br><br>
            <input type="checkbox" id="knew" name="knew" value="News">
            <label for="knew">News</label><br><br>
            <input type="checkbox" id="knew" name="knew" value="Social Media">
            <label for="knew">Social Media</label><br><br>
            <input type="checkbox" id="knew" name="knew" value="Students">
            <label for="knew">Students</label><br><br>
            <input type="checkbox" id="knew" name="knew" value="Websites">
            <label for="knew">Websites</label>
        </div><br>
        <div class="form-group">
            <label for="name">If Other (Please Let Us Know)</label>
            <input type="text" name="name" id="name" placeholder="You learned about
us from...">
        </div>
           <div class="form-group">
             <label for="message">Message</label>
             <textarea name="message" id="message" placeholder="Write
Here..."></textarea>
           </div>
           <button type="submit" class="btn">Submit</button>
           <button type="reset" class="btn">Reset</button>
          </form>
        </div>
      </section>

    <footer class="footer">
      <div class="social">
        <a href="https://www.instagram.com/_soumen_24._/" target="_blank"><i
class="fab fa-instagram fa-2x"></i></a>
        <a href="https://twitter.com/SoumenS86561967" target="_blank"><i class="fab
fa-twitter fa-2x"></i></a>
        <a href="#"><i class="fab fa-youtube fa-2x"></i></a>
        <a href="https://www.linkedin.com/in/soumen-samanta-91866b1b3/"
target="_blank"><i class="fab fa-linkedin fa-2x"></i></a>
      </div>
      <p>Copyright &copy; 2022 - MOI</p>
    </footer>
</body>
</html>
```



MOI

- Home
- Monuments
- Quiz
- Contact

Keep In Touch

Please fill out the form below to contact us

Full Name My Name is...

Email My Email ID is...

How Did You Get To Know About Us?

□ News

□ Social Media

□ Students

□ Websites

If Other (Please Let Us Know) You learned about us from..

Message Write Here...

Copyright © 2022 - MOI

Questions:

1. **What is the use of multiple in list and datalist element?**
   Ans.
   The multiple attribute is a boolean attribute.
   When present, it specifies that multiple options can be selected at once.
   Selecting multiple options vary in different operating systems and browsers:
   • For windows: Hold down the control (ctrl) button to select multiple options
   • For Mac: Hold down the command button to select multiple options
   Because of the different ways of doing this, and because you have to inform the user that multiple selection is available, it is more user-friendly to use checkboxes instead.
   The <datalist> tag specifies a list of pre-defined options for an <input> element.
   The <datalist> tag is used to provide an "autocomplete" feature for <input> elements.
   Users will see a drop-down list of predefined options as they input data.
   The <datalist> element's id attribute must be equal to the <input> element's list attribute (this binds them together).

2. What is the importance of pattern attribute?

The pattern attribute is only applicable on the input element. It allows us to define our own rule to validate the input value using Regular Expressions (RegEx).

• Again, if the value does not match the specified pattern, the input will throw an error.

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

Ans.

1. submit — Submits the current form data. (This is default.)
2. reset — Resets data in the current form.
3. button — Just a button. Its effects must be controlled by something else (that is, with JavaScript).

It used to be the case that buttons primarily appeared in the context of forms, where the default submit behavior made perfect sense. Today, it is common to see buttons in all sorts of in-browser app contexts, and so it isn't always clear at first glance at the markup. Therefore, it is a good practice to always declare the type of a button explicitly.

Outcomes:

Conclusion:
(Conclusion to be based on the outcomes achieved)

Thus we have studied how to create form which

accepts data from the user in html 5.0

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