

EXPERIMENT NO. 02

Aim: Using Tables, Forms and Media.

Markup Language: HTML.

Theory:

(a) Forms and Controls:

The <form> HTML element represents a document section containing interactive controls for submitting information.

input

Description: This form control enables free-form data entry.

```
<input ref="order/shipTo/street" class="streetAddress">
  <caption>Street</caption>
  <hint>Please enter the number and street name</hint>
</input>
```

textarea

Description: This form control enables free-form data entry and is intended for use in entering multiline content, e.g., the body of an email message.

Email Message Body

```
<textarea ref="message/body" class="messageBody">
  <caption>Message Body</caption>
  <hint>Enter the text of your message here</hint>
</textarea>
```

upload

Description: This form control enables the common feature found on Web sites to upload a file from the local file system, as well as accepting

input from various devices including microphones, pens, and digital cameras.

Uploading An Image

```
<upload ref="mail/attach1" mediaType="image/*">  
  <caption>Select image:</caption>  
</upload>
```

button

Description: This form control is similar to the HTML element of the same name and allows for user-triggered actions. This form control may also be used to advantage in realizing other custom form controls.

Simple Button

```
<button>  
  <caption>Click here</caption>  
</button>
```

Submit

```
<submit submitInfo="timecard">  
  <caption>Submit Timecard</caption>  
</submit>
```

Typically, a stylesheet would be used to determine the exact appearance of form controls, though a means is provided to suggest an appearance through attribute selectUI. The value of the attribute consists of one of the following values, each of which may have a platform-specific look and feel.

radio

checkbox

menu

listbox

(b) Demonstrating different semantics

The <section> tag defines a section in a document.

The <nav> tag defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

The <footer> tag defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

(c) Tables

The <table> HTML element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.

(d) Multimedia Features

The <iframe> HTML element represents a nested browsing context, embedding another HTML page into the current one.

Each embedded browsing context has its own session history and document. The browsing context that embeds the others is called the parent browsing context. The topmost browsing context — the one with no parent — is usually the browser window, represented by the Window object.

Sample Code:

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Form with Controls</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Welcome! Sign up to our blog right here!</h1>
```

```

<div>
  <form>
    <div>
      <label for="username">Enter your full name : </label>
      <input type="text" id="username">
    </div>
    <br>
    <div>
      <label for="email">Enter your email id : </label>
      <input type="email" id="email">
    </div>
    <br>
    <div>
      <label for="checkbox">Subscribe to our mailing list</label>
      <input type="checkbox" name="checkbox" id="checkbox">
    </div>
    <br>
    <div>
      <select name="gender" id="gender-selector">
        <option>Male</option>
        <option>Female</option>
        <option>Other</option>
        <option>Refuse to disclose</option>
      </select>
    </div>
    <br>
    <div>
      <h3>Where did you find out about us?</h3>
      <div>
        <div>
          <input type="radio" id="insta" name="type" value="inst
a">

          <label for="insta">Instagram</label>
        </div>

        <div>
          <input type="radio" id="fb" name="type" value="fb">
          <label for="fb">Facebook</label>
        </div>

        <div>
          <input type="radio" id="oa" name="type" value="oa" che
cked>

          <label for="oa">Online Advertisements</label>
        </div>
      </div>
    </div>
  </div>

```

```

        <br>
        <div>
            <label for="opinion">What about our blog appeals to you?</lab
el>
        </div>
        <div>
            <textarea name="opinion" id="opinion" cols="60" rows="10"></te
xtarea>
        </div>
        <br>
    </form>
    <div>
        <a href="page2.html"><button>Subscribe!</button></a>
    </div>

</div>
<footer>
    Made by nickname8888
</footer>
    <script src="" async defer></script>
</body>
</html>

```

Page1.html

```

<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <nav>
            |||<a href="index.html">Login Page</a></li> |||
            <a href="page2.html">Blogs</a></li> |
            <a href="page3.html">Students</a> |
            <a href="page4.html">Videos </a> |
        </nav>
        <h1>Welcome to our homepage of Blogs!</h1>

        <br>
        <section>
            <h3>Intro to Flutter</h3>
            <p>In general, developing a mobile application is a complex and ch
allenging task. There are many frameworks available to develop a mobile applic
ation. Android provides a native framework based on Java language and iOS prov
ides a native framework based on Objective-C / Swift language.

```

However, to develop an application supporting both the OSs, we need to code in two different languages using two different frameworks. To help overcome this complexity, there exist mobile frameworks supporting both OS. These frameworks range from simple HTML based hybrid mobile application framework (which uses HTML for User Interface and JavaScript for application logic) to complex language specific framework (which do the heavy lifting of converting code to native code). Irrespective of their simplicity or complexity, these frameworks always have many disadvantages, one of the main drawback being their slow performance.

In this scenario, Flutter – a simple and high performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native framework.

Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML.

To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and new) and render the widget with only the necessary changes instead of re-rendering the whole widget.

We shall discuss the complete architecture in the coming chapters.

</section>

<section>

<h3>Intro to web design</h3>

<p>This site is a companion for the Introduction to Web Design course. The four main topics we will cover are HTML5 (creating the content for your site), CSS3 (styling your site), JavaScript (adding some interactive features), and Responsive Design (making sure your site looks good on smartphones, tablets, laptops, and PCs). The focus is on best practices and accessibility.

The best way to learn web design is with hands-on learning, so while this site looks good (at least I hope so), it is possible for you to see what the site looked like through each of the four steps.

Just the HTML

Adding the CSS, but just for the large screen view

Before the addition of an interactive picture gallery.</p>

</section>

<section>

Intro to Machine learning

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that “it gives computers the ability to learn without being explicitly programmed”.

And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .”

Machine Learning is the latest buzzword floating around. It deserves to, as it is one of the most interesting subfields of Computer Science. So what does Machine Learning really mean?

Let’s try to understand Machine Learning in layman’s terms. Consider you are trying to toss a paper into a dustbin.

After the first attempt, you realize that you have put too much force into it. After the second attempt, you realize you are closer to the target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned to offer a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing’s proposal in his paper “Computing Machinery and Intelligence”, in which the question “Can machines think?” is replaced with the question “Can machines do what we (as thinking entities) can do?”

Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to “produce reliable, repeatable decisions and results” and uncover “hidden insights” through learning from historical relationships and trends in the data set(input).

Suppose that you decide to check out that offer for a vacation. You browse through the travel agency website and search for a hotel. When you look at a specific hotel, just below the hotel description there is a section titled “You might also like these hotels”. This is a common use case of Machine Learning called “Recommendation Engine”. Again, many data points were used to train a model in order to predict what will be the best hotels to show you under that section, based on a lot of information they already know about you.

So if you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully “learned”, it will then do better at predicting future traffic patterns (performance measure P).

The highly complex nature of many real-world problems, though, often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible. Examples of machine learning problems include, “Is this cancer?”, “Which of these people are good friends with each other?”, “Will this person like this movie?” such problems are excellent targets for Machine Learning, and in fact, machine learning has been applied to such problems with great success. </p>

</section>

<footer>

Made by nickname8888

</footer>

<script src="" async defer></script>

</body>

</html>

Page2.html

<!DOCTYPE html>

<html>

<head>

<title>Students</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<nav>

|||Login Page |||

Blogs |

Students |

Videos |

</nav>

<h1>Students</h1>

<h3>This is the list of Students that contribute to our blogs</h3>

<table>

<thead>

<tr>

<th colspan="2">Students</th>

</tr>

</thead>

<tbody>

<tr>

<td>


```

        <ul>
            <li>Rahul</li>
            <li>Nishant</li>
            <li>Neeraj</li>
            <li>Mahesh</li>
            <li>Harpreet</li>
        </ul>
    </td>
    <td>
        <ul>
            <li>50 Blogs</li>
            <li>12 Blogs</li>
            <li>88 blogs</li>
            <li>3 Blogs</li>
            <li>17 Blogs</li>
        </ul>
    </td>
</tr>
</tbody>
</table>

    <script src="" async defer></script>
</body>
<footer>
    Made by nickname8888
</footer>
</html>

```

Page3.html

```

<!DOCTYPE html>

<html>
    <head>
        <title>Videos</title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <nav>
            |||<a href="index.html">Login Page</a></li> |||
            <a href="page2.html">Blogs</a></li> |
            <a href="page3.html">Students</a> |
            <a href="page4.html">Videos </a> |
        </nav>

        <h1>Videos</h1>
        <h3>Here are all of our videos</h3>

```

```

    <div>
        <iframe width="500" height="300" src="https://www.youtube.com/embed/ZtyMdRzvi0w" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
        <iframe width="500" height="300" src="https://www.youtube.com/embed/4duqI8WyfqE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
        <iframe width="500" height="300" src="https://www.youtube.com/embed/Tn6-PIqc4UM" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
        <iframe width="500" height="300" src="https://www.youtube.com/embed/Ata9cSC2WpM" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
    </div>
    <footer>
        Made by nickname8888
    </footer>
    <script src="" async defer></script>
</body>
</html>

```

Style.css

```

body {
    max-width: fit-content;
    margin: auto;
    background-color: darkcyan;
    color: white;
    margin: 2%;
}

h3 {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

table,
td {
    border: 1px solid #333;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
}

```

```

}

h1 {
    font-family: Arial, Helvetica, sans-serif;
}

thead,
tfoot {
    background-color: #333;
    color: #fff;
}

form {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

nav {
    background-color: cadetblue;
    width: fit-content;
    font-size: large;
}

a {
    color: white;
}

section {
    /* color: black; */
    padding: 1%;
    margin: 5%;
    font-family: Arial, Helvetica, sans-serif;
    font-size: large;
    border: 2px dotted white;
}

footer {
    /* lines to center */
    max-width: fit-content;
    margin: auto;
    margin-top: 10%;
    font-
family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans',
    Arial, sans-serif;
}

iframe {
    margin: 2%;
    padding: 1%;
}

```

```
}    border: 2px dotted white;
```

Output:

Welcome! Sign up to our blog right here!

Enter your full name :

Enter your email id :

Subscribe to our mailing list ☐

▼

Where did you find out about us?

- ☐ Instagram
- ☐ Facebook
- ☐ Online Advertisements

What about our blog appeals to you?

Made by nickname8888

[Login Page](#) | [Blogs](#) | [Students](#) | [Videos](#)

Welcome to our homepage of Blogs!

Intro to Flutter

In general, developing a mobile application is a complex and challenging task. There are many frameworks available to develop a mobile application. Android provides a native framework based on Java language and iOS provides a native framework based on Objective-C / Swift language. However, to develop an application supporting both the OSs, we need to code in two different languages using two different frameworks. To help overcome this complexity, there exists mobile frameworks supporting both OS. These frameworks range from simple HTML based hybrid mobile application framework (which uses HTML for User Interface and JavaScript for application logic) to complex language specific framework (which do the heavy lifting of converting code to native code). Irrespective of their simplicity or complexity, these frameworks always have many disadvantages, one of the main drawback being their slow performance. In this scenario, Flutter – a simple and high performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native framework. Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML. To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and new) and render the widget with only the necessary changes instead of re-rendering the whole widget. We shall discuss the complete architecture in the coming chapters.

Intro to web design

Students

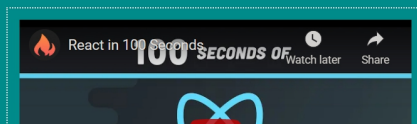
This is the list of Students that contribute to our blogs

Students	
<ul style="list-style-type: none">• Rahul• Nishant• Neeraj• Mahesh• Harpreet	<ul style="list-style-type: none">• 50 Blogs• 12 Blogs• 88 blogs• 3 Blogs• 17 Blogs

Made by nickname8888

Videos

Here are all of our videos



Conclusion:

So we can conclude that we learned about the Tables, Forms and Media its properties