MANIPAL INSTITUTE OF TECHNOLOGY

Manipal – 576 104

DEPARTMENT OF COMPUTER SCIENCE & ENGG.

CERTIFICATE

This is to certify that Ms./Mr. …………………...……………………………………

Reg. No. …..…………………… Section: …………… Roll No.: ………………...

has satisfactorily completed the lab exercises prescribed for Internet Technologies Lab [CSE 4111] of Fourth Year B. Tech. Degree in Computer Science and Engg. at MIT, Manipal, in the academic year 2018–2019.

Date: ……....................................

Signature
Faculty in Charge

CONTENTS

Course Objectives

- Acquire in-depth understanding of web application architecture.
- To understand techniques to improve user experience in web applications.
- To gain knowledge about how to interact with database, files and XML.

Course Outcomes

At the end of this course, students will have the

- Ability to develop a basic website using a modern web development tool.
- Ability to design websites with better look and feel.
- Expertise to create real-world web applications.

Evaluation Plan

- Internal Assessment Marks : 60%
  - ✓ Continuous evaluation component (for each experiment):10 marks
  - ✓ The assessment will depend on punctuality, program execution, maintaining the observation note and answering the questions in viva voce.
  - ✓ The marks of the 10 experiments except the mini project is valued out of 40 marks.
  - ✓ The mini project is valued out of 20 marks.
  - ✓ The total lab internal marks is 60.

- End semester assessment of 2 hour duration: 40 %

INSTRUCTIONS TO THE STUDENTS

Pre-Lab Session Instructions

1. Students should carry the Lab Manual Book and the required stationery to every lab session.
2. Be in time and follow the institution dress code.
3. Must Sign in the log register provided.
4. Make sure to occupy the allotted seat and answer the attendance.
5. Adhere to the rules and maintain the decorum.

In-Lab Session Instructions

- Follow the instructions on the allotted exercises.
- Show the program and results to the instructors on completion of experiments.
- On receiving approval from the instructor, copy the program and results in the Lab Record.
- Prescribed textbooks and class notes can be kept ready for reference if required.

General Instructions for the exercises in Lab

- Implement the given exercise individually and not in a group.
- The programs should meet the following criteria:
    - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
    - Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
    - Comments should be used to give the statement of the problem.
    - Statements within the program should be properly indented.
    - Use meaningful names for variables and functions.

  - Make use of constants and type definitions wherever needed.
  - The website should be well designed and unique.
- Plagiarism (copying from others) is strictly prohibited and would invite severe penalty in evaluation.
- The exercises for each week are divided under three sets:
  - Solved exercise
  - Lab exercises – to be completed during lab hours.
  - Additional Exercises – to be completed outside the lab or in the lab to enhance the skill.
- In case a student misses a lab class, he/she must ensure that the experiment is completed during the repetition lab with the permission of the faculty concerned but credit will be given only to one day’s experiment(s).
- Questions for lab tests and examination are not necessarily limited to the questions in the manual, but may involve some variations and / or combinations of the questions.
- A sample on how to write the lab observation is provided with this lab note.

THE STUDENTS SHOULD NOT

- Bring mobile phones or any other electronic gadgets to the lab.
- Go out of the lab without permission.

LAB NO.: 1 **Date.**

HTML5, CSS AND JAVASCRIPT

Objectives:

In this lab, student will be able to:

- Develop HTML5 web pages
- Familiarize with Cascading Style Sheets
- Embed the JavaScript code in HTML5 pages

I. DESCRIPTION

HTML5 – Hyper Text Markup Language Version 5

HTML5 is the 5th and newest version of HTML standard, providing new features like rich media support, interactive web applications etc.

The most interesting HTML5 elements are:

1. Semantic elements like <header>, <footer>, <article> and <section>
2. Attributes of form elements like number, date, time, calendar and range.
3. Graphic elements like <svg> and <canvas>
4. Multimedia elements like <audio> and <video>

There are several Application Programming Interfaces too in HTML5 like HTML Geolocation, HTML Drag and Drop, HTML Web Workers etc.

Several elements of HTML4 have been removed in HTML5 like <big>, <center>, <font>, <frame>, <frameset>, <strike> etc.

To indicate that your HTML content uses HTML5, simply add <!DOCTYPE html> on top of the html code.

Procedure to create and HTML document:

In notepad type the necessary code & save with the file name mentioned with .html or .htm extension.

Example:

```
<html>
<head>
<title> My First Page </title>
</head>
```

```
<body>
<h1> Hello </h1>
<h2> Welcome to Internet Technologies Lab </h2>
</body>
</html>
```

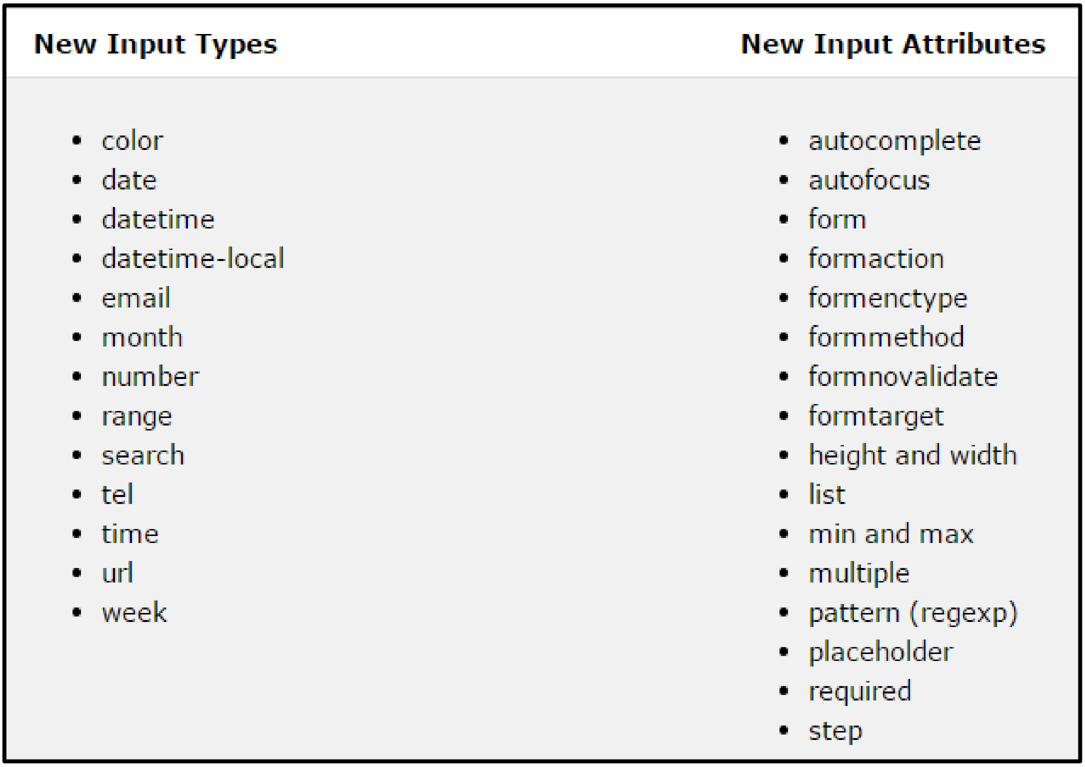
HTML5 Elements

HTML5 offers new elements for better document structure. The below given table gives a brief description on few HTML5 elements.

TAG	**DESCRIPTION**
<article>	Defines an article in a document
<dialog>	Defines a dialog box or window
<header>	Defines a header for a document or a section
<footer>	Defines a footer for a document or a section
<nav>	Defines navigation links
<time>	Defines a date/time
<output>	Defines the result of a calculation
<canvas>	Draw graphics, on the fly, via scripting(JavaScript)
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements
<video>	Defines video or movie

Figure 1.1 Few HTML5 elements

HTML5 Input types and Attributes



New Input Types

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New Input Attributes

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Figure 1.2 HTML5 input types and attributes

The above figure lists the new input types and attributes of HTML5.

HTML5 Events

On visiting a website the user perform actions like clicking on links or image, hover over things etc. These are considered to be examples for Events.

Event handlers are developed to handle these events and this can be done using a scripting language like JavaScript, VBScript etc wherein event handlers are specified as a value of event tag attribute.

The following attributes (very few) can be used to trigger any **javascript** or **vbscript** code given as value, when there is any event occurs for any HTM5 element.

Attribute	**Value**	**Description**

offline	script	Triggers when the document goes offline
onchange	script	Triggers when an element changes
onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondrag	script	Triggers when an element is dragged
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes
onload	script	Triggers when the document loads
onmousedown	script	Triggers when a mouse button is pressed
onpause	script	Triggers when a media data is paused
onselect	script	Triggers when an element is selected
onsubmit	script	Triggers when a form is submitted

Figure 1.3 HTML5 event attributes

HTML5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. The user must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content. The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an Id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Example: To draw a circle

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
 </body>
</html>
```

Output

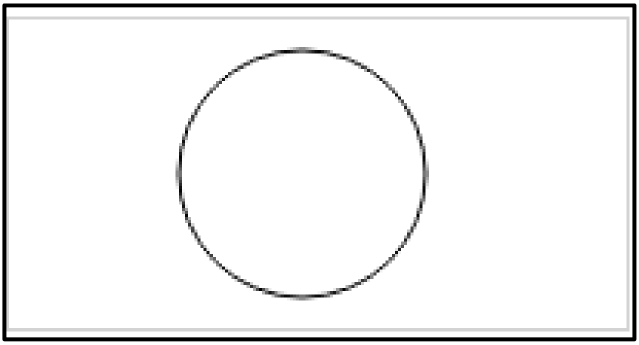


Figure 1.4

HTML5 Web Forms

The HTML <form> element defines a form that is used to collect user input. Form elements consist of various input elements like text field, check boxes, radio buttons, submit buttons etc.

```
<form>

</form>
```

Input Element

It is the most important form element and can be displayed in several ways, depending on the type attribute.

<input type = “text”>   Defines a one line text input field

<input type = “radio”>   Defines a radio button

<input type = “submit”>   Defines a submit button

Action attribute

The action attribute defines the action to be performed when the form is submitted. Usually the form data is sent to a web page on the server when the user clicks on the submit button.

<form action = “/action_page.aspx”>

// action_page.aspx contains a server side script that handles the form data.

If the action attribute is omitted, the action is set to the current page.

Method attribute

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.

<form action = “/action_page.aspx” method=”get”>

The default method when submitting form data is GET. When GET is used the submitted form data will be visible in the page address field. Therefore it must not be used when sending sensitive information.

Use POST method if the form data contains sensitive or personal information. It does not display the submitted form data in the page address field. It has no size limitations and can be used to send large amounts of data.

CSS – Cascading Style Sheet

CSS is a stylesheet language used for describing the presentation of a document written in a markup language ie it describes the style of a web document including the layout, design and display variations for various displays. CSS can be applied to a web document in 3 ways.

1) Inline style – Right next to the text it decorates, by using style attribute.
   <h1 style = “color : blue ;”> Hello </h1>

2) Internal style – At the top of the web page document, using <style> element in <head>

```
<head>

<style>

h1  { color : blue ;}

</style> </head>
```

3) External style – in a separate file

```
<head>

<link rel=”stylesheet” href = “style.css”>

</head>
```

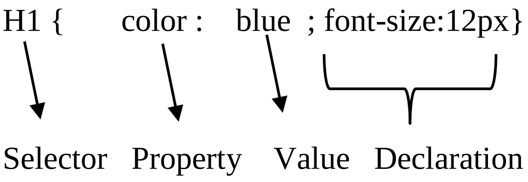
style.css

```
h1  { color : blue ;}
```

The style definitions are usually saved in an external stylesheet since changing one single file can help in redesigning the entire web document with new look and feel.

CSS syntax

A CSS rule set consists of a selector and a declaration block. The selector points to the HTML element to be styled. The declaration block contains one or more declarations separated by semicolons.



H1 {   color :   blue ; font-size:12px}

Selector   Property   Value   Declaration

CSS Selectors are used to “find” or select HTML elements based on their element name, id, class, attribute etc. The element selector selects the elements based on the element name. The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page. To select an element with a specific id, write a # character followed by the id of the element.

```
#para1{

 text-align: center;

color:red; }
```

The class selector selects the elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
.center {

text-align: center;

color:red; }
```

JavaScript

JavaScript is a light weight and interpreted programming language. It is a scripting language that is commonly used for the client side web development. It makes the HTML pages more dynamic and interactive.

It can be used to put dynamic text in to HTML page (Eg: document.write("<h1>" + name + "</h1>"); // write variable text in to HTML page), react to events, validate data, create cookies etc.

Syntax

It can be implemented using JavaScript statements that are placed within the <script>….</script> HTML tags in a web page. The <script> tags, containing the JavaScript code can be placed anywhere within the web page, but normally recommended to place within the <head> tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as script. The script tag have mainly two attributes language and type, specifying the scripting language used.

To select an HTML element, JavaScript very often use the document.getElementById(id) method. The word document.write is a standard JavaScript command for writing output to a page.

```
<script language=”javascript” type=”text/javascript”>

document.getElementById("demo").innerHTML = "Hello JavaScript!";

</script>
```

II. SOLVED EXERCISE:

Develop an HTML5 program to validate the credentials with appropriate internal styling with the help of CSS and JavaScript.

Program:

<u>Login.html</u>

```
<!DOCTYPE html>
<html>
<head>
<title>  Login page  </title>
<style>
body {
   background-color: lightblue;
}

h1 {
   color: white;
   text-align: center;
}

p {
   font-family: verdana;
   font-size: 20px;
}
</style>
<script language="javascript">
function check(form)       /*function to check userid & password*/
{
  /*the following code checkes whether the entered userid and password are
matching*/
 if(form.userid.value == "myuserid" && form.pswrd.value == "mypswrd")
  {
    window.open("https://www.google.com", "_blank");
/*opens the target page while Id & password matches*/
  }
 else
 {
   alert("Error Password or Username")    /*displays error message*/
  }
}
</script>
</head>
<body>
<h1> Validate Credentials </h1>
<form name="login">
Username<input type="text" name="userid"/>
Password<input type="password" name="pswrd"/>
<input type="button" onclick="check(this.form)" value="Login"/>
<input type="reset" value="Cancel"/>
</form>
</body>
</html>
```

Output

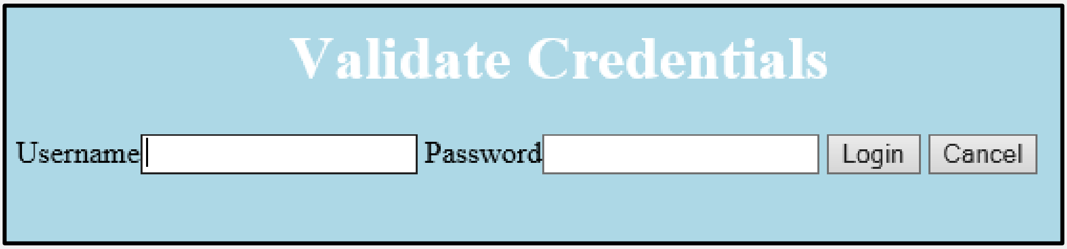


Figure 1.5

III. LAB EXERCISE:

1) Create an HMTL5 document to get an HTML5 element's position on the web page with the help of CSS and JavaScript function.
2) Write a JavaScript program to Wish a user at different hours of a day. Use appropriate dialog boxes for wishing the user. Display the dynamic clock on the web page. Make use of CSS and HTML5 elements for creative and attractive designs.
3) Create an HTML5 document that displays a bouncing ball. Use HTML5 elements, CSS and JavaScript functions.
4) Design a form that includes all the following.
   Form related tags with all the possible attributes: <INPUT> with all TYPEs, <SELECT>, <TEXTAREA> etc. On clicking the submit button a message "Mr.X has booked number of tickets from source_name to destination_name" must be displayed. The age of the person cannot be less than 25 and the maximum number of tickets that can be booked is 20.

ADDITIONAL EXERCISES:

1) Develop a color-picker using HTML5 elements, CSS and JavaScript functions.
2) Create an animation of rain using HTML5 canvas element. Apply appropriate usage of CSS and JavaScript function to develop the animation.