Ex. No. 1 **Date:**	**WEBSITE CREATION**	**Page No.:**

Aim:

To create a website for the department using HTML, CSS and JavaScript.

Programming Fundamentals:

HTML BASICS:

Basic HTML

Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Defines information about the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph
<br>	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment

Formatting

Tag	Description
<acronym>	Not supported in HTML5. Use <abbr> instead. Defines an acronym
<abbr>	Defines an abbreviation or an acronym
<address>	Defines contact information for the author/owner of a document/article
<b>	Defines bold text
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
<center>	Not supported in HTML5. Use CSS instead. Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<del>	Defines text that has been deleted from a document
<dfn>	Represents the defining instance of a term
<em>	Defines emphasized text
<font>	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<i>	Defines a part of text in an alternate voice or mood
<ins>	Defines a text that has been inserted into a document
<kbd>	Defines keyboard input

<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<pre>	Defines preformatted text
<progress>	Represents the progress of a task
<q>	Defines a short quotation
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<small>	Defines smaller text
<strike>	Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text
<strong>	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<template>	Defines a template
<time>	Defines a date/time
<tt>	Not supported in HTML5. Use CSS instead. Defines teletype text
<u>	Defines text that should be stylistically different from normal text
<var>	Defines a variable
<wbr>	Defines a possible line-break

Forms and Input

Tag	**Description**
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<button>	Defines a clickable button
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

Frames

Tag	**Description**
<frame>	Not supported in HTML5.

	Defines a window (a frame) in a frameset
<frameset>	Not supported in HTML5. Defines a set of frames
<noframes>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<iframe>	Defines an inline frame

Images

Tag	Description
<img>	Defines an image
<map>	Defines a client-side image-map
<area>	Defines an area inside an image-map
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content
<picture>	Defines a container for multiple image resources
<svg>	Defines a container for SVG graphics

Audio / Video

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements (<video>, <audio> and <picture>)
<track>	Defines text tracks for media elements (<video> and <audio>)
<video>	Defines a video or movie

Links

Tag	Description
<a>	Defines a hyperlink
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<nav>	Defines navigation links

Lists

Tag	Description
<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dir>	Not supported in HTML5. Use <ul> instead. Defines a directory list
<dl>	Defines a description list
<dt>	Defines a term/name in a description list
<dd>	Defines a description of a term/name in a description list

Tables

Tag	Description
<table>	Defines a table
<caption>	Defines a table caption
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table
<col>	Specifies column properties for each column within a <colgroup> element
<colgroup>	Specifies a group of one or more columns in a table for formatting

Styles and Semantics

Tag	Description
<style>	Defines style information for a document
<div>	Defines a section in a document
<span>	Defines a section in a document
<header>	Defines a header for a document or section
<footer>	Defines a footer for a document or section
<main>	Specifies the main content of a document
<section>	Defines a section in a document
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<summary>	Defines a visible heading for a <details> element
<data>	Links the given content with a machine-readable translation

Meta Info

Tag	Description
<head>	Defines information about the document
<meta>	Defines metadata about an HTML document
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document

Programming

Tag	Description
<script>	Defines a client-side script
<noscript>	Defines an alternate content for users that do not support client-side scripts

<applet>	Not supported in HTML5. Use <embed> or <object> instead. Defines an embedded applet
<embed>	Defines a container for an external (non-HTML) application
<object>	Defines an embedded object
<param>	Defines a parameter for an object

Cascading Style Sheet(CSS):

Cascading Style Sheet(CSS) is used to set the style in web pages that contain HTML elements, here we will see in how many ways we can add CSS for our HTML, there three different ways to do so one by one we will see those procedure.

Types:

External CSS: External CSS contains a separate CSS file with a .css extension which contains only style property with the help of tag attributes.

```
selector{
   property1: value1;
   property2: value2;
}
```

Include external CSS file: The external CSS file is linked to the HTML document using a link tag.

```
<link rel="stylesheet" type="text/css" href="/style.css" />
```

Internal CSS or Embedded: CSS is embedded within the HTML file using a style HTML tag.

```
<style type="text/css">
        div { color: #444;}
</style>
```

Inline CSS: It contains CSS properties in the body section specified within HTML tags.

```
<tag style="property: value"> </tag>
```

Selectors: Used to find or select the HTML elements you want to style. These are categorized as follows:

Basic Selectors	Description	Syntax
Universal	Selects all elements on the pages.	*{property:value;}
Type	Selects all HTML tag/element of given type in your document.	p {property:value;}
Id	Selects an element based on the value of its unique id attribute	#id {property:value;}
Class	Selects all elements in the document that have the given class attribute.	.class {property:value;}

Attribute	Selects all elements that have a specified attribute.	a[attribute=value]
		{property:value;}

Font Properties: CSS font properties are used to set the font's content of the HTML element as per requirement.

Property	Description	Syntax
Font-family	Specifies the font family to be used for the element's text content.	font-family: family-name |generic-family |initial |inherit;
Font-style	Styles the text content in a normal, italic, or oblique face from its font-family.	font-style: normal |italic |oblique |initial |inherit;
Font-variant	Converts all lowercase letters into uppercase letters.	font-variant: normal| small caps | initial;
Font-weight	Specifies thickness or weight of the font	font-weight: normal| bold |number |initial |inherit |unset;
Font-size	Specifies the size of the text in HTML document.	font-size: small |medium |large

Text-properties: CSS text formatting properties are used to format and style text by setting their color, alignment, spacing, etc. as per requirement.

Property	Description	Syntax
Text-color	Sets the color of the text.	color: value;
Text-alignment	Defines the horizontal alignment of the text.	text-align: left|right|center| justify|initial|inherit;
Text-decoration	Add or remove text- decorations.	text-decoration: decoration-type;
Text-transformation	Changes the case(uppercase|lowercase) of text.	text-transform: none|capitalize|uppercase| lowercase|initial|inherit;

Text-indentation	Indents the first line of text block.	text-indent: length|initial|inherit;
Letter spacing	Specifies spacing between the characters of the text.	letter-spacing: normal|length|initial|inherit;
Line height	Specifies the space between the lines of the text.	line-height: normal|number|length|percentage|initial|inherit;
Text-shadow	Adds shadow to the text.	text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit
Word spacing	Specifies space between words of lines.	word-spacing: normal|length|initial|inherit;

Background properties: The CSS background properties are used to design the background and define the background effects for elements.

Property	**Description**	**Syntax**
Background-color	Specifies the background color of an element.	background-color: color_name;
Background-image	Adds one or more background images to an element.	background-image: url('url');
Background-repeat	Adds or remove repeat the background image both horizontally and vertically.	background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;
Background-position	Specifies the positioning of the image in a certain way.	background-position: value;
Background-origin	Used to adjust the background image of the webpage.	background-origin: padding-box |border-box |content-box | initial| inherit;

Background-attachment	Specifies the kind of attachment of the background image in its container/td>	background-attachment: scroll|fixed|local|initial|inherit;
Background-clip	Used to define how far the background (color or image) should extend.	background-clip: border-box|padding-box|content-box|initial|inherit;

Box Properties: The CSS box model is essentially a box that wraps around every HTML element consisting of the border, padding, margin, and content. The CSS properties used to attain the box model are:

Property	Description	Syntax
Margin	Used to set the margin	margin: value;
Padding	Specifies the space between the border and the content of the selector.	padding: value;
Border	Sets the element's border width and set the style, and color of an element's border.	border: value;
Width	Used to set an element's width.	width: value;
Height	Used to set an element's height	height: value;

Shadow properties: These shadow properties are used to add shadow to text or boxes or frames of elements.

Property	Description	Syntax
Text shadow	Adds shadow to text.	text-shadow: h-shadow v-shadow blur-radius color| none |initial | inherit;
Box shadow	Gives shadow-like effect to the box or frames of an element.	box-shadow: h-offset v-offset blur spread color |none |inset |initial | inherit;

Gradient: The CSS gradient property is used to create transition between two or more specified colors

Gradient	Description	Syntax
Linear Gradient	Creates smooth color transitions.	background-image: linear-gradient(direction, color-stop1, color-stop2, …);
Radial Gradient	Used to obtain an elliptical shape gradient.	background-image: radial-gradient(shape size at position, start-color, …, last-color);

Border Properties: The CSS border properties allow you to specify how the border of the box representing an element should look.

Property	Description	Syntax
Border Color	Specifies the color of the border of the box. Works only when the border-style property is defined.	border-color: color-value;
Border Style	Sets the style of the border as solid, dotted, rigged, etc.	border-style: value;
Border Width	Sets the width of the border of the element.	border-width: length |thin |medium |thick |initial |inherit

Classification Properties: The CSS classification properties allow you to specify how and where an element is displayed.

Property	Description	Syntax
Display	Defines how elements are displayed in the web page.	display: inline|block|flex| grid|table|group|none|inherit;
Float	Defines flow of content	float: none|left|right| initial| inherit;
Position	Specifies the positioning method of html entity on the web page.	position: fixed|static|absolute |relative |sticky;

Clear	Sets the sides of an element where no other floating elements are allowed.	clear: left|right|both|none;
Visibility	Set an element as visible or not.	visibility: visible|hidden|collapse|initial|inherit;
Cursor	Specifies the type or shape of cursor	cursor: auto|default|pointer|crosshair|help | e-resize | all-scroll |progress |initial |inherit;

CSS Functions: CSS has a range of inbuilt functions. These are used as a value for various CSS properties. Some of the CSS functions can be nested as well. It ranges from simple color functions to mathematical, shape, color, transform, gradient, and animations functions. Some of the key functions are:

Function	**Description**	**Syntax**
attr()	Retrieves the value of an attribute of the selected elements	attr(attr_name);
calc()	Takes a single mathematical expression as its parameter and performs operations	calc(Expression);
max()	Returns the largest number of the given set of comma separated numbers.	max(value 1, value2, value3…)
url()	Takes a string URL as a parameter and is used to load images, fonts and content	url(<string> <url-modifier>*)
var()	Inserts the value of a custom property. Its name must start with two dashes.	var(custom_property, value)

JavaScript Fundamentals

Syntax:

```
<script type = "text/javascript">
        //JavaScript coding can be done inside this tag
</script>
OR
<script src="script.js"></script>
```

Javascript Variables

JavaScript allows the usage of variables in the following three ways:

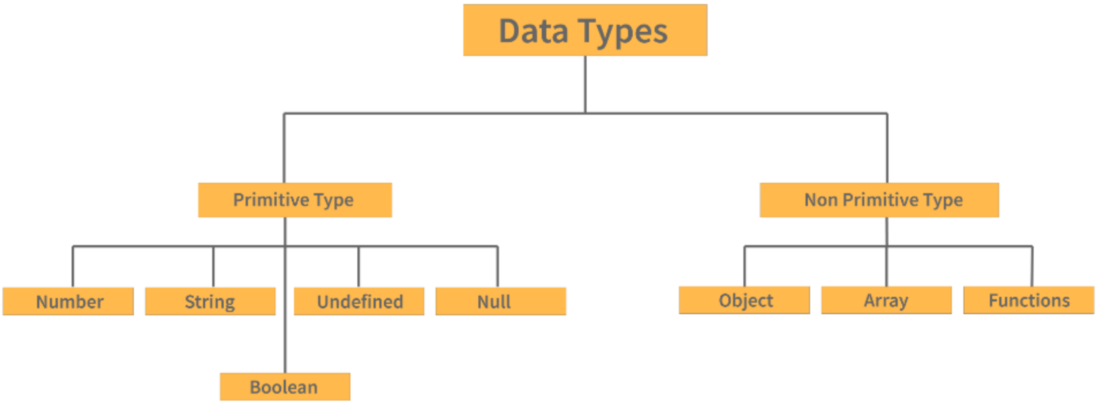
i. var

var x = 140; // variable x can be reassigned a new value and also redeclared

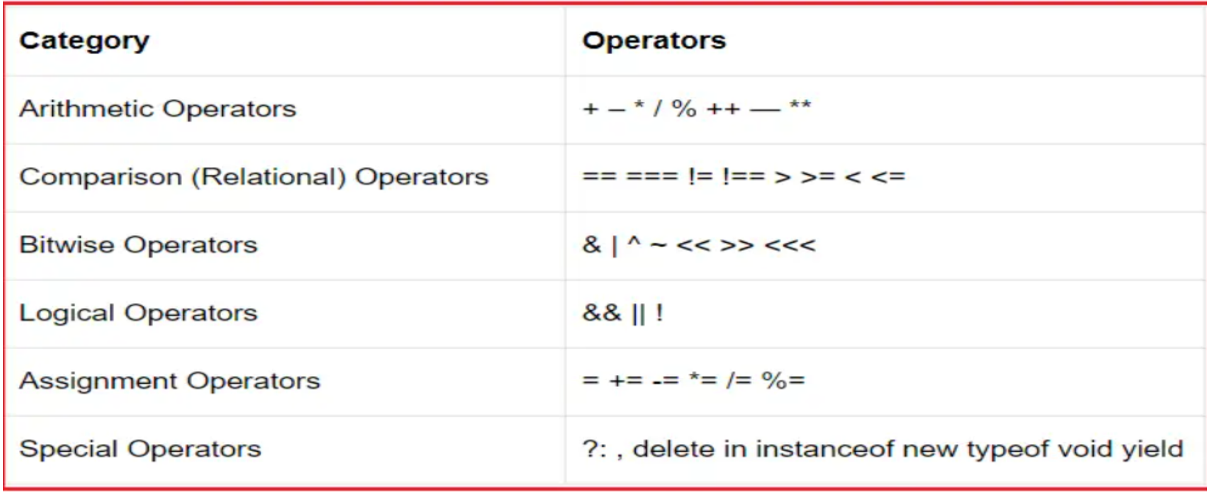
ii. const: They can neither be reassigned values, that is, their value remains fixed throughout the execution of the code, nor can they be redeclared.
const x = 5; // variable x cannot be reassigned a new value or redeclared

iii. let: The let variable, like const, cannot be redeclared. But they can be reassigned a value. An example of a variable declared using the "let" keyword in JavaScript is shown below:
let x = 202; // variable x cannot be redeclared but can be reassigned a new value

Javascript Data Types



Javascript Operators



Category	Operators
Arithmetic Operators	+ – * / % ++ — **
Comparison (Relational) Operators	== === != !== > >= < <=
Bitwise Operators	& | ^ ~ << >> <<<
Logical Operators	&& || !
Assignment Operators	= += -= *= /= %=
Special Operators	?: , delete in instanceof new typeof void yield

JavaScript If-Else Statements

```
if (check condition) {
  // block of code to be executed if the given condition is satisfied
} else {
  // block of code to be executed if the given condition is not satisfied
}
```

Loops in JavaScript:

For loop

for (initialization of the loop variable; condition checking for the loop; updation after the loop) {
  // code to be executed in loop
}

While loop

// Initialization of the loop variable is done before the while loop begins

```
while(condition checking for the loop){
// 1. code to be executed in loop
// 2. updation of the loop variable
}
```

Do-while loop

```
// Initialization of the loop variable is done before the do-while loop begins
do{
// 1. code to be executed in loop
// 2. updation of the loop variable
}while(condition checking for the loop);
```

the continue statement: Skip parts of the loop if certain conditions are met.
break statement: Used to stop and exit the cycle when specific conditions are met.

JavaScript Arrays

Arrays are the next item on our JavaScript cheat sheet. Arrays are used in a variety of programming languages. They are a method of categorising variables and attributes. Arrays can be defined as a collection of objects of the same type. In JavaScript, here's how one can make an array of cars:

```
var cars = ["Mercedes", "Tesla","Volvo"];
```

- pop(): This method is used for removing the last element of an array.
- push(): This method is used for adding a new element at the very end of an array.
- concat(): This method is used for joining various arrays into a single array.
- reverse(): This method is used for reversing the order of the elements in an array.
- shift(): This method is used for removing the first element of an array.
- slice(): This method is used for pulling a copy of a part of an array into a new array.
- splice(): This method is used for adding elements in a particular way and position.
- toString(): This method is used for converting the array elements into strings.
- unshift(): This method is used for adding new elements at the beginning of the array.
- valueOf(): This method is used for returning the primitive value of the given object.
- indexOf(): This method is used for returning the first index at which a given element is found in an array.
- lastIndexOf(): This method is used for returning the final index at which a given element appears in an array.
- join(): This method is used for combining elements of an array into one single string and then returning it.
- sort(): This method is used for sorting the array elements based on some condition.

JavaScript Functions

JavaScript Functions can be defined as chunks of code written in JavaScript to perform a single task. A function in JavaScript looks like this:

```
function nameOfTheFunction(parameters) {
  // Job or Task of the function
}
```

Coding:

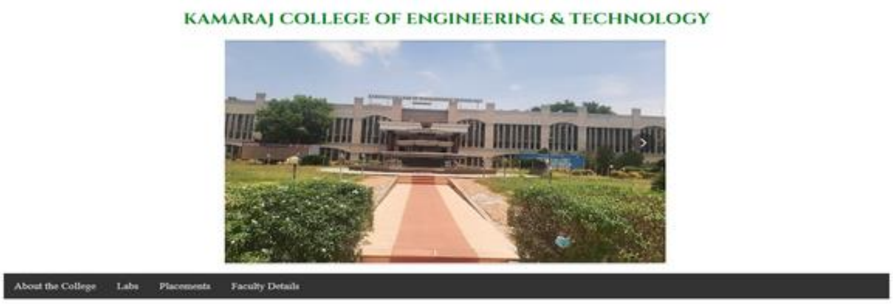
```
<html>
<head>
<link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css"/>
<link
href="https://fonts.googleapis.com/css?family=Cinzel:400,700,900|Tangerine|Anton|Libre+B
askerville|Lobster" rel="stylesheet"  type="text/css">
<style>
.bk{
  background-image: ("http://bit.ly/2ydhsHj");
}
 .img-responsive{

  }
.green-text{
    font-family: Cinzel;
    color: green;
      }
  .red-text {
    color: red;
    font-family:Anton;
    text-align:justify;
  }
.pstyle
{
  font-size:28px;
  }

h2 {
    font-family: Lobster, Monospace;
  }

.bl-text{
  font-family:LibreBaskerville;
  color:black;
  font-size:20px;
  text-align:justify;
}
.bul-text{
  font-family:Tangerine;
  color:black;
  font-size=30px;
  text-align:justify;
}
.ls-text{
  list-style-image: url("http://bit.ly/2ApzHuw");
  font-size:20px;
  font-family="Lobster"
  text-align="justify"
  }
```

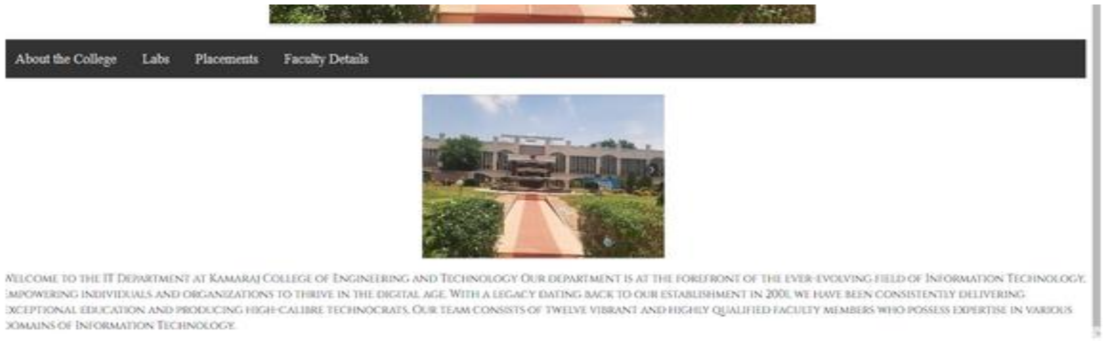
```
.d-text{
  font-family:Lobster;
  font-size:20px;
  text-align:justify;
}

.topnav {
    background-color: #333;
    overflow: hidden;
}

.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 17px;
}

.topnav a:hover {
    background-color: #ddd;
    color: black;
}

.topnav .icon {
    display: none;
}
@media screen and (max-width: 600px) {
  .topnav a:not(:first-child) {display: none;}
  .topnav a.icon {
    float: right;
    display: block;
  }
}

@media screen and (max-width: 600px) {
  .topnav.responsive {position: relative;}
  .topnav.responsive a.icon {
    position: absolute;
    right: 0;
    top: 0;
  }
  .topnav.responsive a {
    float: none;
    display: block;
    text-align: left;
  }
}
```

```
img:hover {
    -webkit-transform: scaleX(-1);
    transform: scaleX(-1);
}

.target{
   font-family: Cinzel;
  font-size:14px;
    color: black;
  text-align=justify;
}
.target >div{
  display:none;
}
.target >div:target{
display:block;
  }
.map-responsive{
    overflow:hidden;
    padding-bottom:56.25%;
    position:relative;
    height:0;
}
.map-responsive iframe{
    left:0;
    top:0;
    height:100%;
    width:100%;
    position:absolute;
}
</style>
<style>
function myFunction() {
    var x = document.getElementById("myTopnav");
    if (x.className === "topnav") {
        x.className += " responsive";
    } else {
        x.className = "topnav";
    }
}
</style></head>
<body class="bk">
  <h1 class="green-text pstyle"><p align=center> KAMARAJ COLLEGE OF
ENGINEERING & TECHNOLOGY</p></h1>
<p align="center"><img src="applelab.jpg" alt="Kcet" ></p>
<div class="topnav" id="myTopnav">
  <a href="#About">About the College</a>
  <a href="#Infra">Labs</a>
    <a href="#Placements">Placements</a>
   <a href="#faculty">Faculty Details</a> </div>
```

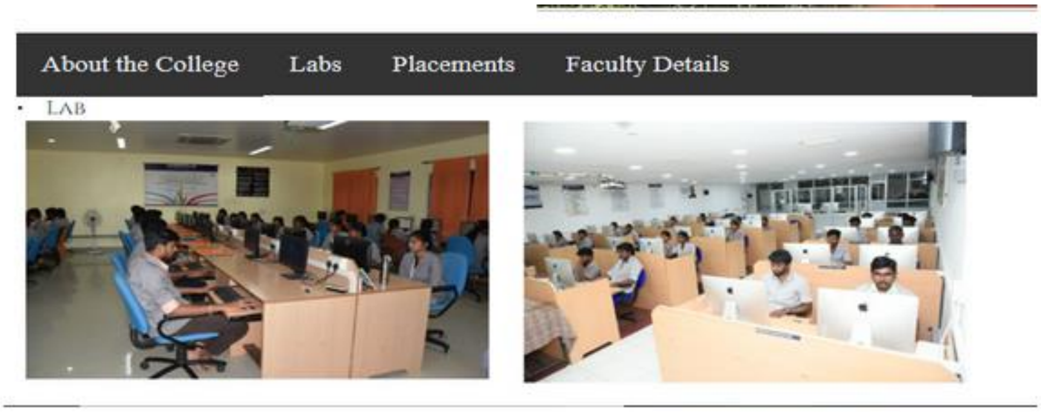
```
<div class="target"><div id="About"><p class="bl-text">
<center><img src="college.jpg" alt="college" width=300px height=200px></center></p>
<p>Welcome to the IT Department at Kamaraj College of Engineering and Technology Our
department is at the forefront of the ever-evolving field of Information Technology,
empowering individuals and organizations to thrive in the digital age. With a legacy dating
back to our establishment in 2001, we have been consistently delivering exceptional
education and producing high-calibre technocrats. Our team consists of twelve vibrant and
highly qualified faculty members who possess expertise in various domains of Information
Technology.
</p>
</div>
<div id="Infra">
    <li>Lab</li>
   <img src="D:\Pictures\Saved Pictures\seclab.jpg" alt="KGCAS" width=300px
height=200px>
   <img src="D:\Pictures\Saved Pictures\applelab.jpg" alt="KGCAS" width=300px
height=200px>
  </div>

  <div id="Placements">
<image src="plc.JPG">
  </div>
  <div id="faculty">
<img src="pic1.PNG"alt="pic1">
<img src="pic2.PNG"alt="pic2">
<img src="pic3.PNG"alt="pic3">
<img src="pic4.PNG"alt="pic4">/<br>
<img src="pic5.PNG"alt="pic5">
   </div>
      </div>
</body>
</html>
```

Output:

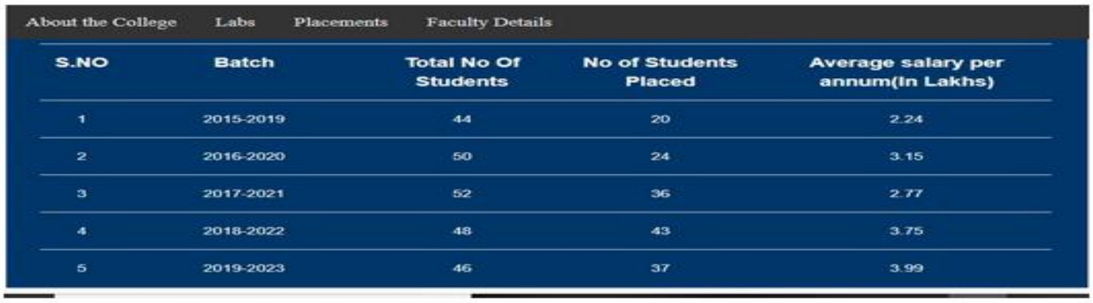




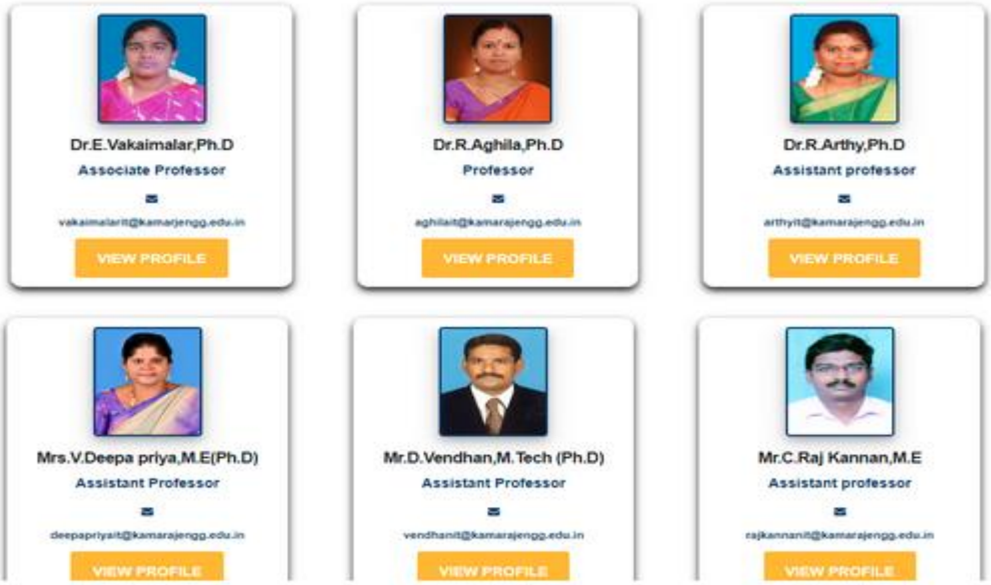
Welcome to the IT Department at Kamaraj College of Engineering and Technology Our department is at the forefront of the ever-evolving field of Information Technology, empowering individuals and organizations to thrive in the digital age. With a legacy dating back to our establishment in 2001, we have been consistently delivering exceptional education and producing high-calibre technocrats. Our team consists of twelve vibrant and highly qualified faculty members who possess expertise in various domains of Information Technology.



- Lab



S.NO	Batch	Total No Of Students	No of Students Placed	Average salary per annum(In Lakhs)
1	2015-2019	44	20	2.24
2	2016-2020	50	24	3.15
3	2017-2021	52	36	2.77
4	2018-2022	48	43	3.75
5	2019-2023	46	37	3.99



Dr.E.Vakaimalar,Ph.D
Associate Professor
✉
vakaimalarit@kamarjengg.edu.in

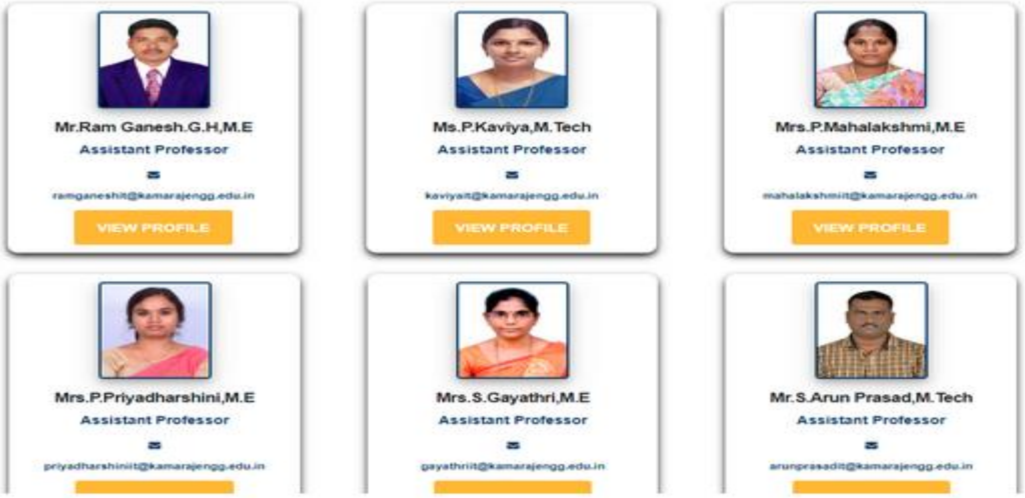
Dr.R.Aghila,Ph.D
Professor
✉
aghilait@kamarajengg.edu.in

Dr.R.Arthy,Ph.D
Assistant professor
✉
arthyit@kamarajengg.edu.in

Mrs.V.Deepa priya,M.E(Ph.D)
Assistant Professor
✉
deepapriyait@kamarajengg.edu.in

Mr.D.Vendhan,M.Tech (Ph.D)
Assistant Professor
✉
vendhanit@kamarajengg.edu.in

Mr.C.Raj Kannan,M.E
Assistant professor
✉
rajkannanit@kamarajengg.edu.in



Mr.Ram Ganesh.G.H,M.E
Assistant Professor
✉
ramganeshit@kamarajengg.edu.in

Ms.P.Kaviya,M.Tech
Assistant Professor
✉
kaviyait@kamarajengg.edu.in

Mrs.P.Mahalakshmi,M.E
Assistant Professor
✉
mahalakshmiit@kamarajengg.edu.in

Mrs.P.Priyadharshini,M.E
Assistant Professor
✉
priyadharshiniit@kamarajengg.edu.in

Mrs.S.Gayathri,M.E
Assistant Professor
✉
gayathriit@kamarajengg.edu.in

Mr.S.Arun Prasad,M.Tech
Assistant Professor
✉
arunprasadit@kamarajengg.edu.in

Result:

Thus the concepts of HTML, CSS and JavaScript are used to create the department website.

Rubrics	
User Interface [20]	
Compilation [10]	
Validation [20]	
Viva [25]	
Record Submission [25]	
Total [100]	

Viva Questions:

HTML:

1. What does HTML stand for? Explain its purpose in web development.
2. How do you create a hyperlink in HTML?
3. What is the difference between inline and block-level elements in HTML?
4. How do you add comments in HTML?
5. Explain the usage of the <img> tag and how you can specify alternate text for an image.
6. What are semantic elements in HTML? Provide examples of a few semantic tags.
7. How can you create a table in HTML? Provide an example.
8. What is the purpose of the <!DOCTYPE> declaration in an HTML document?

9. Describe the difference between the id and class attributes in HTML.
10. Create the following table:

Day	Session	Agenda
Mon	FN	Language Training
	AN	Aptitude
Tues	FN	Skill Rack
	AN	Programming Training
Wed	FN	Group Discussion
	AN	Public Speaking

CSS:

11. What does CSS stand for? Why is it important in web development?
12. How do you link an external CSS file to an HTML document?
13. Explain the usage of the box model in CSS.
14. What is the difference between padding and margin in CSS?
15. How can you apply CSS styles to specific HTML elements based on their class or ID?
16. Describe the difference between inline, block, and inline-block display properties in CSS.
17. How do you create a CSS animation? Provide an example.
18. Explain the concept of CSS specificity and how it affects the application of styles.
19. What is a media query in CSS? How does it help in making a website responsive?
20. Apply animations to a button using CSS, so it changes color gradually when hovered over and returns to its original color when the mouse pointer moves away.

JavaScript:

21. What is JavaScript, and what are its main features?
22. How do you declare a variable in JavaScript?
23. Explain the difference between let, const, and var in variable declaration.
24. How can you check the data type of a variable in JavaScript?
25. What are the different ways to create a function in JavaScript?
26. How do you add an event listener to an HTML element in JavaScript?
27. Explain the concept of scope in JavaScript.
28. What is the purpose of the ‘this’ keyword in JavaScript?
29. How can you handle errors in JavaScript using try-catch blocks?
30. Implement a JavaScript countdown timer that starts from a specified time and updates every second until it reaches zero.