A REPORT OF ONE MONTH TRAINING

at

HB INFOTECH SOLUTIONS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEEGRE OF

BACHELOR OF TECHNOLOGY

Computer Science and Engineering

JUNE-JULY 2024

SUBMITTED BY :

NAME: JIYA GARG

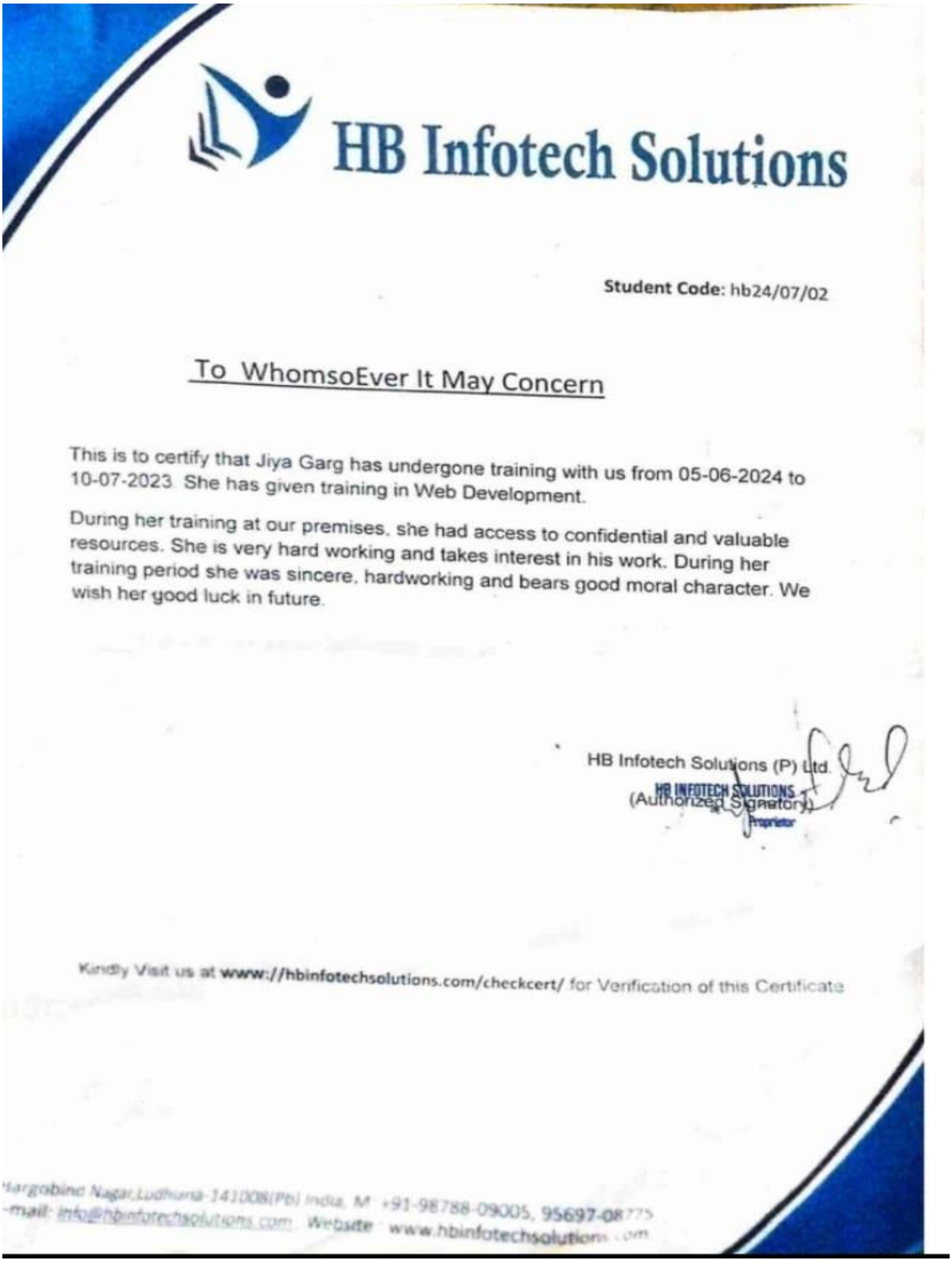
UNIVERSITY ROLL NO: 2104131

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE LUDHIANA

(An Autonomous College Under UGC ACT)

CERTIFICATE BY COMPANY



HB Infotech Solutions

Student Code: hb24/07/02

To WhomsoEver It May Concern

This is to certify that Jiya Garg has undergone training with us from 05-06-2024 to 10-07-2023. She has given training in Web Development.

During her training at our premises, she had access to confidential and valuable resources. She is very hard working and takes interest in his work. During her training period she was sincere, hardworking and bears good moral character. We wish her good luck in future.

HB Infotech Solutions (P) Ltd.
(Authorized Signatory)

HB INFOTECH SOLUTIONS
Proprietor

Kindly Visit us at www://hbinfotechsolutions.com/checkcert/ for Verification of this Certificate

Hargobind Nagar,Ludhiana-141008(Pb) India, M: +91-98788-09005, 95697-08775
-mail: info@hbinfotechsolutions.com Website: www.hbinfotechsolutions.com

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

CANDIDATE'S DECLARATION

I “JIYA GARG” hereby declare that I have undertaken one month training at “**Hb Infotech Solutions**” during a period from June 5, 2024 to July 5, 2024 in partial fulfilment of requirements for the award of degree of B-Tech (Computer Science Engineering) at Guru Nanak Dev Engineering College, Ludhiana. The work which is being presented in the training report submitted to Department of Computer Science Engineering at Guru Nanak Dev Engineering College, Ludhiana is an authentic record of training work.

Signature of the Student

The one-month industrial training Viva–Voce Examination of _________________ has been held on ____________ and accepted.

Signature of Internal Examiner   Signature of External Examiner

ABSTRACT

This training program on web development provided participants with a comprehensive overview of the essential skills and technologies required to build and maintain modern websites. Spanning several weeks, the curriculum covered key topics including HTML, CSS, JavaScript, and responsive design principles. Participants engaged in hands-on projects that facilitated practical application of theoretical knowledge, allowing them to create functional web pages and user interfaces. Additionally, the training introduced frameworks and libraries such as Bootstrap, emphasizing best practices for code organization and user experience. By the end of the program, attendees had developed a solid foundation in web development, equipping them with the skills to pursue further learning. This training aimed not only to enhance technical skills but also to foster critical thinking and problem-solving abilities essential in the rapidly evolving field of web development.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who contributed to the success of this training program. First and foremost, I extend my sincere thanks to the trainer Mr Manpreet Singh and mentors of HB Infotech Solutions for their invaluable guidance, expertise, and support throughout the course. Their dedication and passion for web development inspired and motivated me to push my boundaries. The discussions, shared experiences, and teamwork greatly enriched the learning process. Their commitment to fostering professional development is truly commendable.

This training has been a significant stepping stone in my career, and I am grateful to everyone who played a role in making it a success.

ABOUT THE COMPANY

HB Infotech Solutions is a leading software services company known for its commitment to delivering world-class integrated business solutions tailored for the IT industry. They specialize in IT consulting services and web application development, consistently striving for top performance through the use of cutting-edge technology, high-quality research, and rigorous quality processes.

Their marketing department is comprised of experts focused on accelerating business growth and innovation. HB Infotech Solutions offers a wide range of specialized software services, ensuring exceptional support and comprehensive access to the best market opportunities.

Their main areas of focus include Quality Consulting, Software Testing, and Outsourcing, all underscored by a commitment to quality and client satisfaction. HB Infotech Solutions provides high quality Web Application Services that all aspects of your software application life-cycle needs. Our support team is efficient enough to handle and deliver the clients run time requirement at any cost. Customized Software Solutions HB Infotech Solutions is dedicated to develop flexible, custom-designed software meeting all the requirements of the customers has helped us earn an unmatched reputation among the customers. The company prioritizes excellence, making it a point to deliver unparalleled quality in every project.

LIST OF FIGURES

LIST OF TABLES

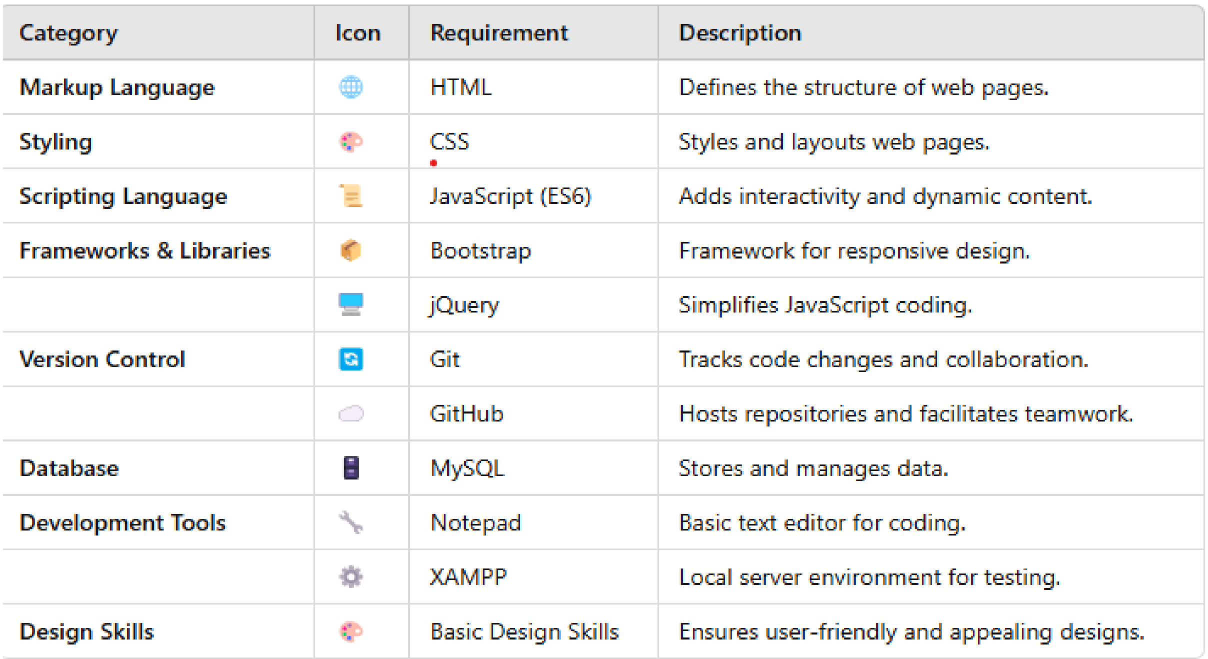
CONTENTS

Topic **Page No.**

CHAPTER-1 INTRODUCTION

1.1 WEB DEVELOPMENT

Web development is the work involved in developing a web site for the Internet or an intranet. Web development can range from developing a simple single static page of plain text to complex web-based internet applications, electronic businesses, and social network service. A website is a collection of related web pages, including multimedia content, typically identified by a common domain name and published on at least one web server. A website may be accessible via a public Internet Protocol (IP) network, such as the Internet, or a private local area network (LAN), by referencing a uniform resource locator (URL) that identifies the site. All publicly accessible websites collectively constitute the World Wide Web, while private websites, such as a company's website for its employees, are typically part of an intranet.



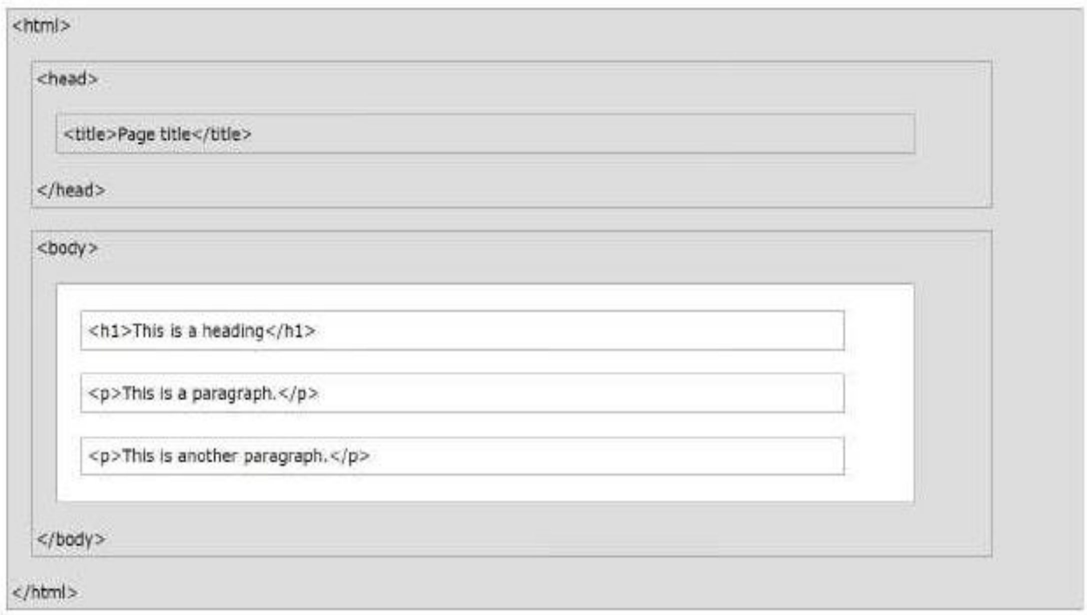
Category	Icon	Requirement	Description
Markup Language	🌐	HTML	Defines the structure of web pages.
Styling	🎨	CSS	Styles and layouts web pages.
Scripting Language	📜	JavaScript (ES6)	Adds interactivity and dynamic content.
Frameworks & Libraries	📦	Bootstrap	Framework for responsive design.
	🖥️	jQuery	Simplifies JavaScript coding.
Version Control	🔄	Git	Tracks code changes and collaboration.
	☁️	GitHub	Hosts repositories and facilitates teamwork.
Database	🗄️	MySQL	Stores and manages data.
Development Tools	🔧	Notepad	Basic text editor for coding.
	⚙️	XAMPP	Local server environment for testing.
Design Skills	🎨	Basic Design Skills	Ensures user-friendly and appealing designs.

Figure 1.1

1.2 HTML

HTML stands for Hyper Text Markup Language HTML is the standard markup language for Web pages HTML elements are the building blocks of HTML pages HTML elements are represented by tags BASIC TERMS:

1.2.1 PROJECT STRUCTURE:



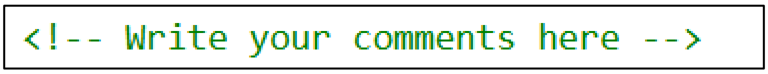
```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

Figure 1.1

- The <html> element is the root element of an HTML page
- The <head> element contains header related tags information such as ‘title for the page’ and ‘links for the CSS files’
- The <title> element specifies a title for the HTML page
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

1.2.2 COMMENTS IN HTML

Comments are not displayed by the browser, but they can help document your HTML source code. You can also hide more than one line. Everything between the <!-- and the --> will be hidden from the display.



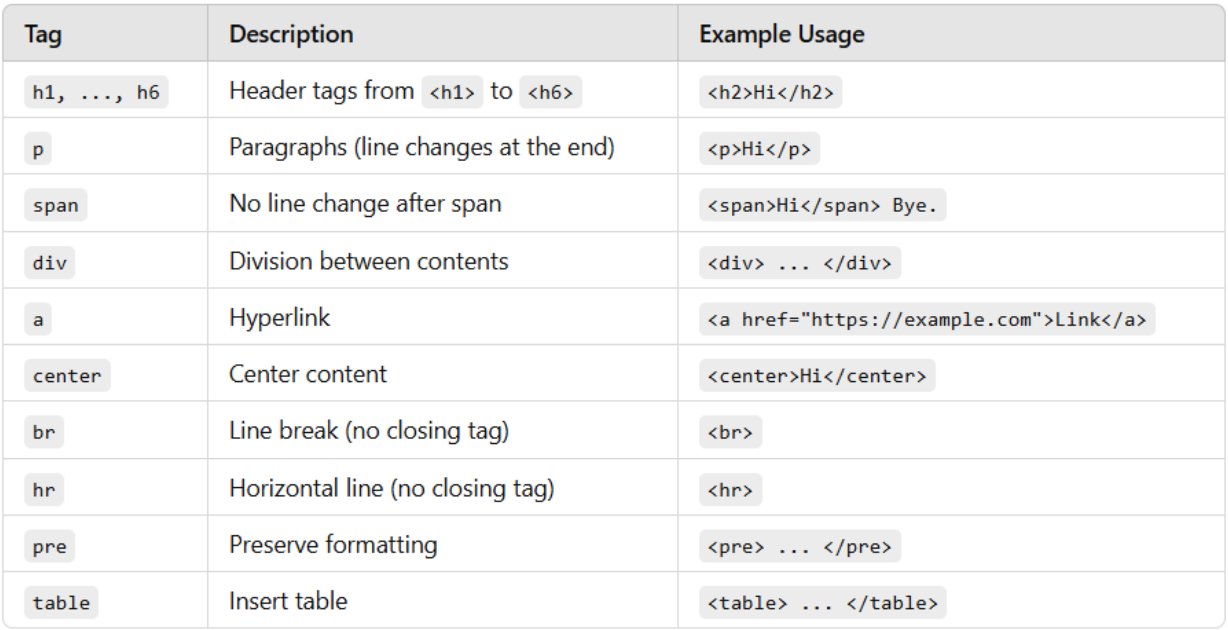
```
<!-- Write your comments here -->
```

Figure 1.2

1.2.3 BASIC TAGS

HTML tags are like keywords which defines that how web browser will format and display the content.

Table 1.1



Tag	Description	Example Usage
`h1, ..., h6`	Header tags from `<h1>` to `<h6>`	`<h2>Hi</h2>`
`p`	Paragraphs (line changes at the end)	`<p>Hi</p>`
`span`	No line change after span	`<span>Hi</span> Bye.`
`div`	Division between contents	`<div> ... </div>`
`a`	Hyperlink	`<a href="https://example.com">Link</a>`
`center`	Center content	`<center>Hi</center>`
`br`	Line break (no closing tag)	`<br>`
`hr`	Horizontal line (no closing tag)	`<hr>`
`pre`	Preserve formatting	`<pre> ... </pre>`
`table`	Insert table	`<table> ... </table>`

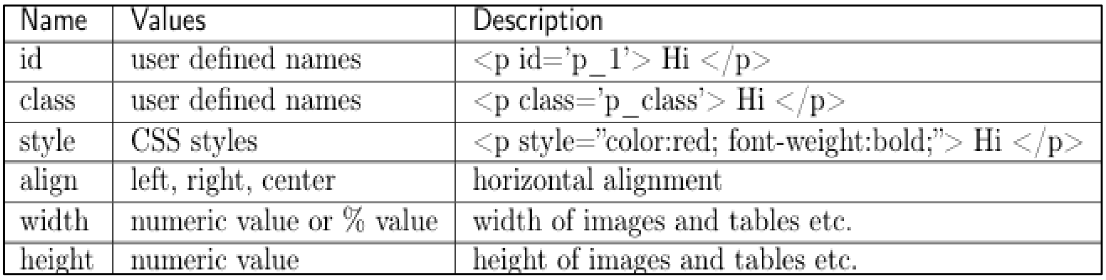
1.2.3 ATTRIBUTES

- HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
- The attributes should always be applied with name and value pair.
- Syntax: <element attribute_name="value">content</element>

Example:

<p style="height: 50px; color: blue" It will add style to element</p>

Table 1. 2

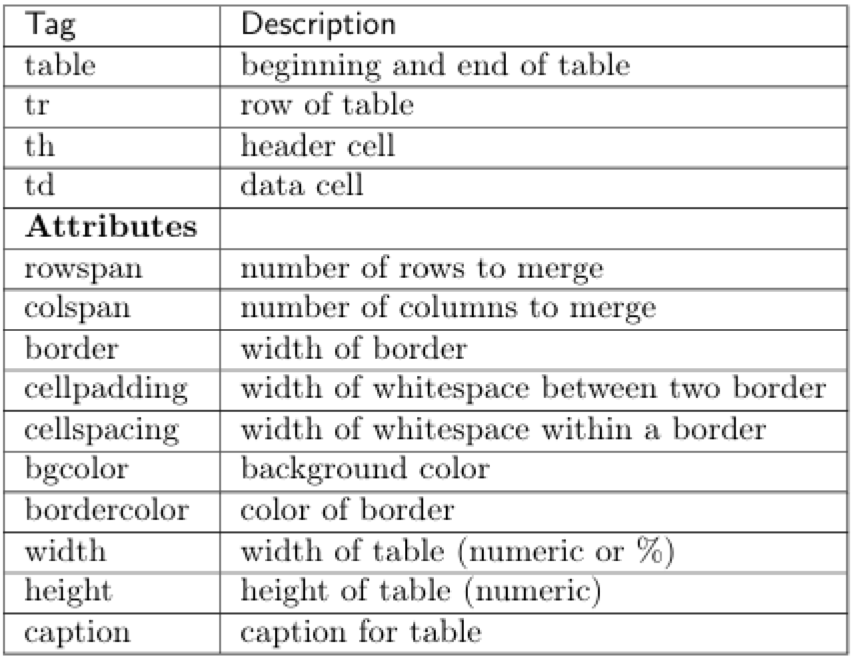


Name	Values	Description
id	user defined names	<p id='p_1'> Hi </p>
class	user defined names	<p class='p_class'> Hi </p>
style	CSS styles	<p style="color:red; font-weight:bold;"> Hi </p>
align	left, right, center	horizontal alignment
width	numeric value or % value	width of images and tables etc.
height	numeric value	height of images and tables etc.

1.2.4 TABLES

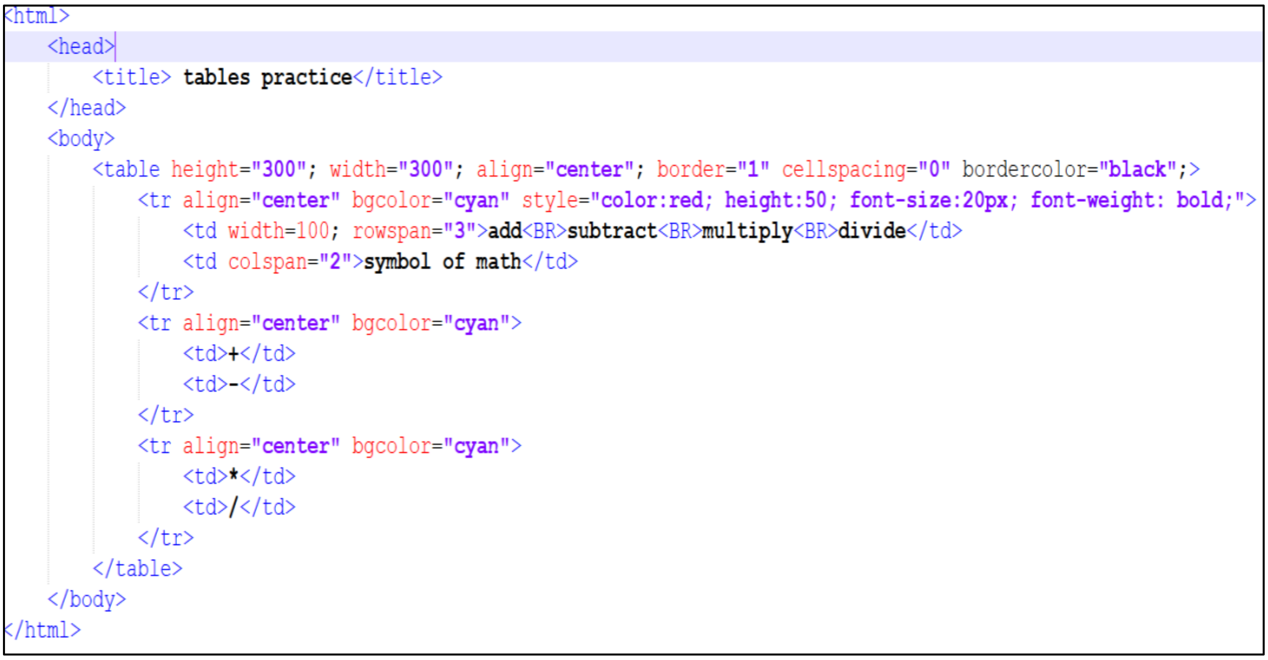
- HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.
- We can create a table to display data in tabular form, using <table> element, with the help of <tr>, <td> and <th> elements.

Table 1.3



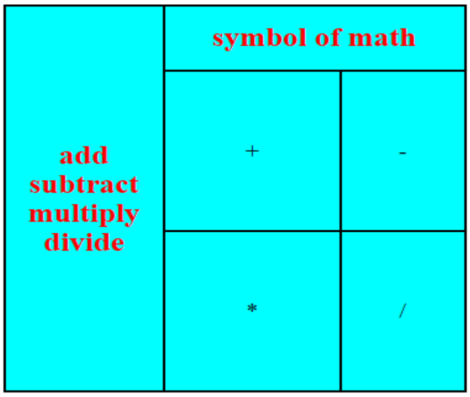
Tag	Description
table	beginning and end of table
tr	row of table
th	header cell
td	data cell
Attributes	
rowspan	number of rows to merge
colspan	number of columns to merge
border	width of border
cellpadding	width of whitespace between two border
cellspacing	width of whitespace within a border
bgcolor	background color
bordercolor	color of border
width	width of table (numeric or %)
height	height of table (numeric)
caption	caption for table

Example :



```
<html>
    <head>
        <title> tables practice</title>
    </head>
    <body>
        <table height="300"; width="300"; align="center"; border="1" cellspacing="0" bordercolor="black";>
            <tr align="center" bgcolor="cyan" style="color:red; height:50; font-size:20px; font-weight: bold;">
                <td width=100; rowspan="3">add<BR>subtract<BR>multiply<BR>divide</td>
                <td colspan="2">symbol of math</td>
            </tr>
            <tr align="center" bgcolor="cyan">
                <td>+</td>
                <td>-</td>
            </tr>
            <tr align="center" bgcolor="cyan">
                <td>*</td>
                <td>/</td>
            </tr>
        </table>
    </body>
</html>
```

Figure 1.3



<table>
<tr><td rowspan="3">add
subtract
multiply
divide</td><td colspan="2">symbol of math</td></tr>
<tr><td>+</td><td>-</td></tr>
<tr><td>*</td><td>/</td></tr>
</table>

Figure 1.4

1.2.5 HTML Images and Text Formatting

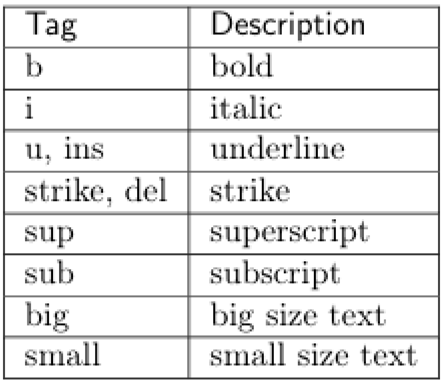
Image tag has two important attributes i.e. ‘src’ and ‘alt’ as described below

src: tells the location of ‘image’ file e.g. the image ‘logo.jpg’ will be searched inside the folder ‘img’ alt: is the ‘alternate text’ which is displayed if image is not found. For example, if the name of the image is incorrectly written i.e. ‘logos’ (instead of ‘logo’), therefore the value of ‘alt’ i.e. ‘Missing Logo.jpg’ will be displayed.

```
<img src=”logo.jpg” alt=” Missing Logo.jpg” height="180" width="300" >
```

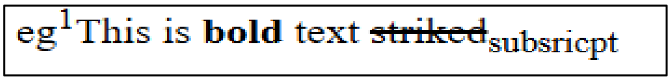
For text formatting these tags are used:

Table 1.4



Tag	Description
b	bold
i	italic
u, ins	underline
strike, del	strike
sup	superscript
sub	subscript
big	big size text
small	small size text

Example : <p>eg<sup>1</sup>This is <b> bold</b> text <strike>striked</strike><sub>subsricpt</sub></p>



eg^{1}This is **bold** text ~~striked~~$_{\text{subsricpt}}$

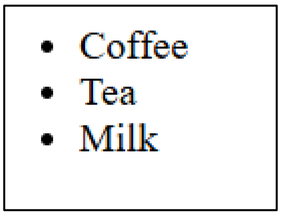
Figure 1. 5

1.2.6 LIST

There are various types of lists in HTML.

Unordered list: bullet is used in it. An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

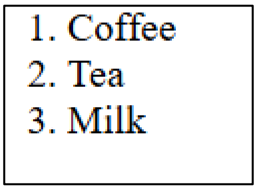


- Coffee
- Tea
- Milk

Figure 1.6

Ordered list: numbers are used in it. An ordered list starts with the <ol> tag. Each list item starts with the <li> tag (Defines a list item).

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



1. Coffee
2. Tea
3. Milk

Figure 1.7

1.2.7 LINKS

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

<a href="*url*">*link text*</a> . By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

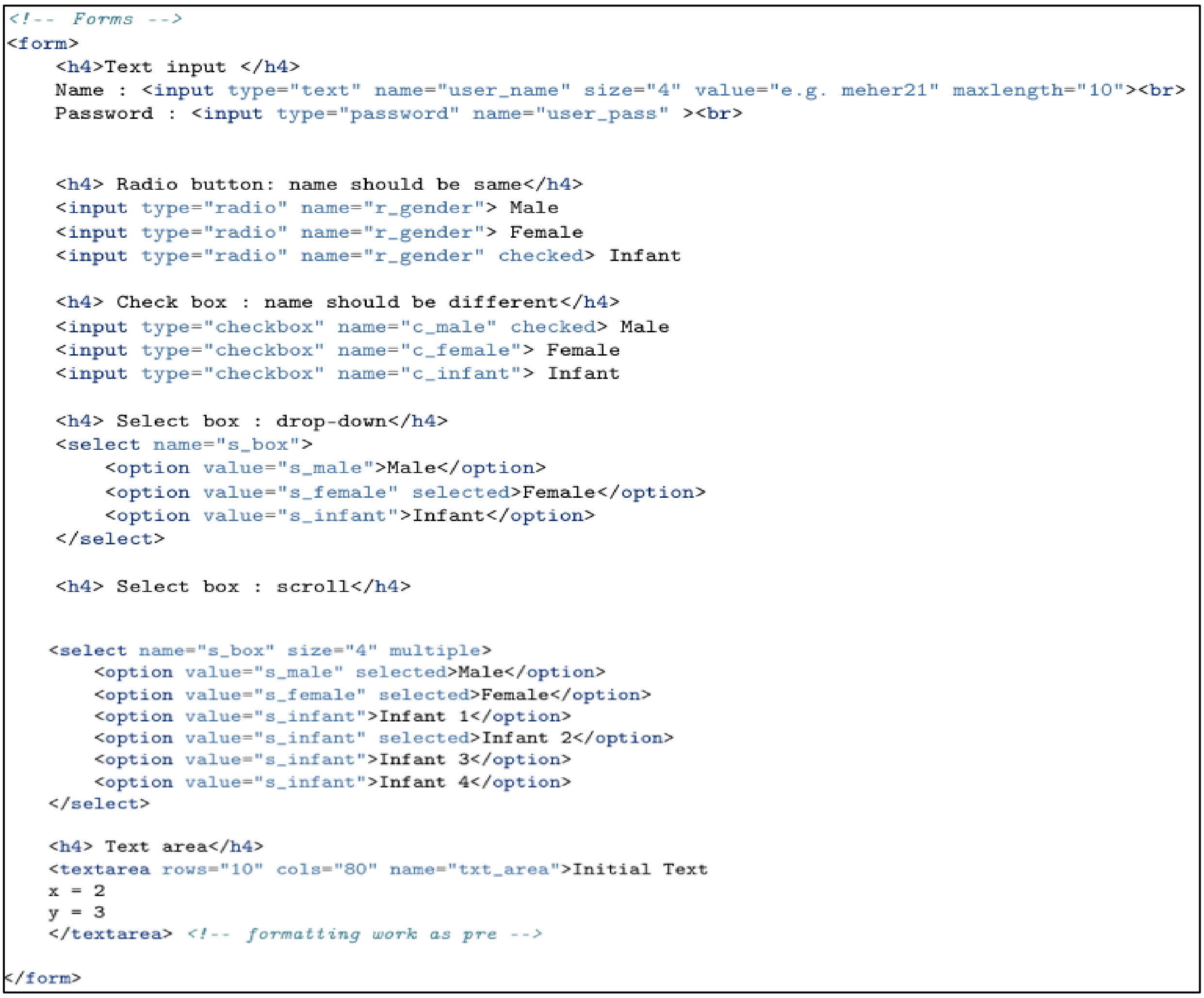
The target attribute specifies where to open the linked document.

<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>

1.2.8 FORM

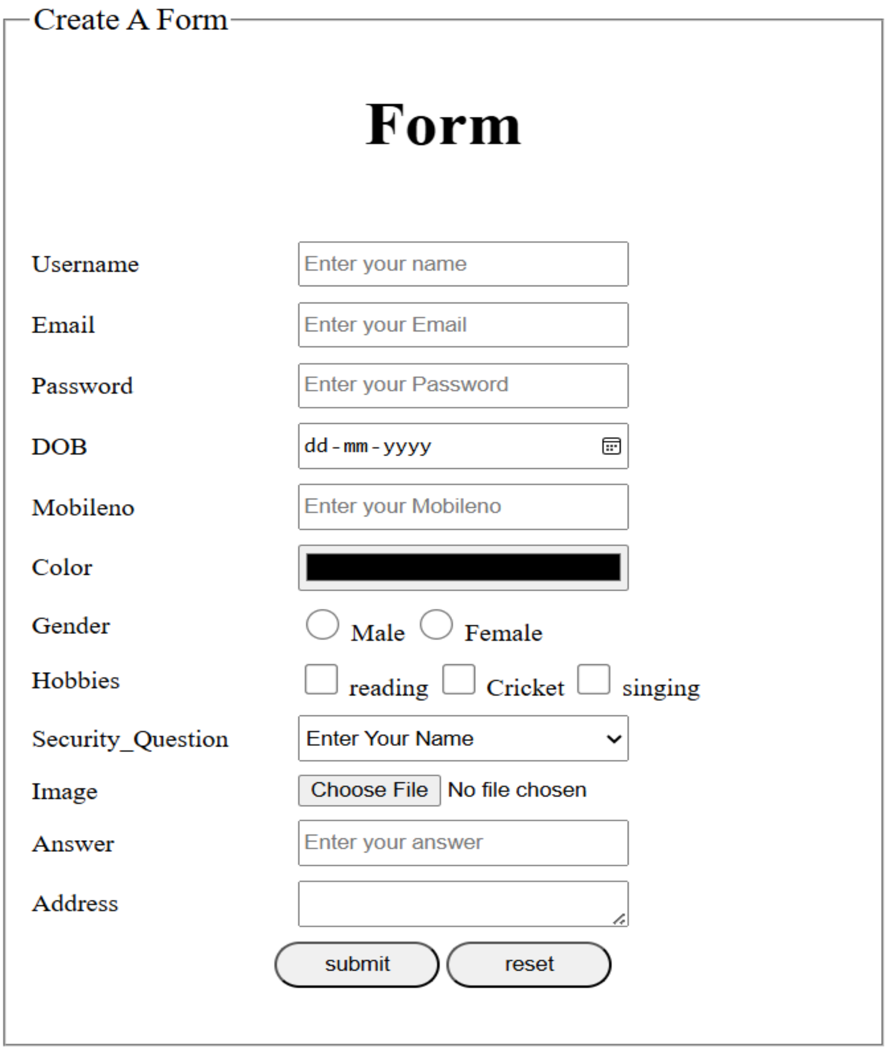
An **HTML form** is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.



```
<!-- Forms -->
<form>
    <h4>Text input </h4>
    Name : <input type="text" name="user_name" size="4" value="e.g. meher21" maxlength="10"><br>
    Password : <input type="password" name="user_pass" ><br>


    <h4> Radio button: name should be same</h4>
    <input type="radio" name="r_gender"> Male
    <input type="radio" name="r_gender"> Female
    <input type="radio" name="r_gender" checked> Infant

    <h4> Check box : name should be different</h4>
    <input type="checkbox" name="c_male" checked> Male
    <input type="checkbox" name="c_female"> Female
    <input type="checkbox" name="c_infant"> Infant

    <h4> Select box : drop-down</h4>
    <select name="s_box">
        <option value="s_male">Male</option>
        <option value="s_female" selected>Female</option>
        <option value="s_infant">Infant</option>
    </select>

    <h4> Select box : scroll</h4>


    <select name="s_box" size="4" multiple>
        <option value="s_male" selected>Male</option>
        <option value="s_female" selected>Female</option>
        <option value="s_infant">Infant 1</option>
        <option value="s_infant" selected>Infant 2</option>
        <option value="s_infant">Infant 3</option>
        <option value="s_infant">Infant 4</option>
    </select>

    <h4> Text area</h4>
    <textarea rows="10" cols="80" name="txt_area">Initial Text
    x = 2
    y = 3
    </textarea> <!-- formatting work as pre -->

</form>
```

Figure 1.8



Create A Form

Form

Username	Enter your name
Email	Enter your Email
Password	Enter your Password
DOB	dd - mm - yyyy
Mobileno	Enter your Mobileno
Color	■
Gender	○ Male ○ Female
Hobbies	□ reading □ Cricket □ singing
Security_Question	Enter Your Name
Image	Choose File No file chosen
Answer	Enter your answer
Address	

submit reset

Figure 1.9

1.2.9 IFRAME

The HTML <iframe> tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document. <iframe src="*url*" title="*description*"></iframe>

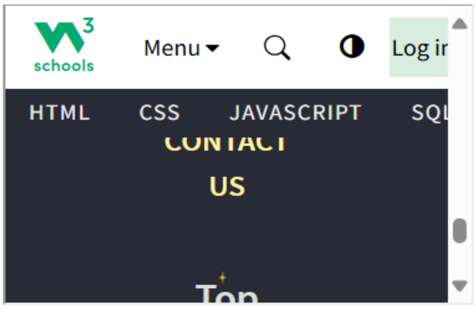


Figure 1.11

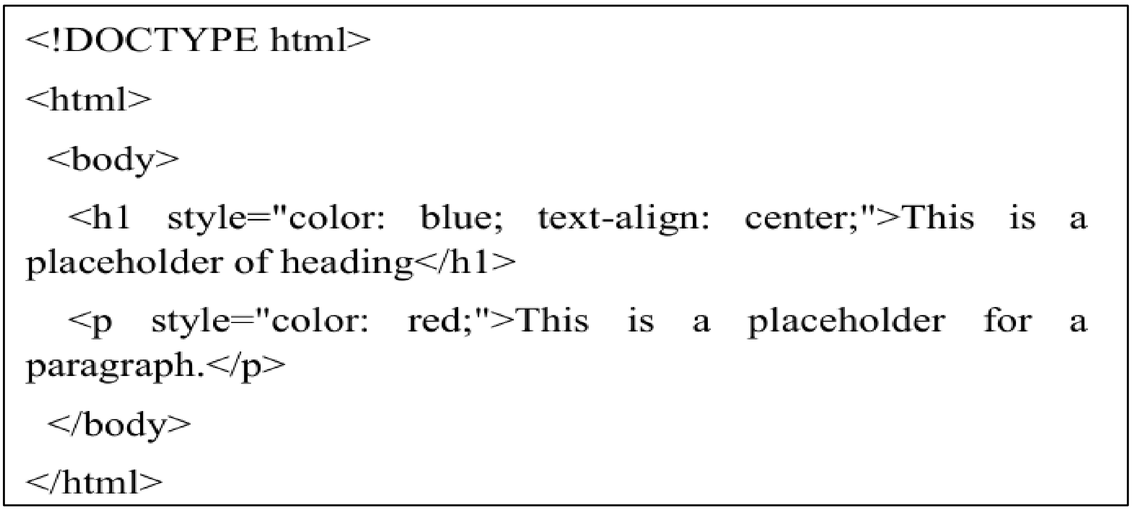
1.3 CSS

CSS (Cascading Style Sheets) is a style sheet language used to control the layout and appearance of HTML documents. It is a powerful tool for styling web pages and enhancing their user experience.

1.3.1 Types of CSS

There are three main types of CSS:

1. **Inline CSS:** This type of CSS is used to style a single HTML element directly within the HTML code. It is less efficient and more challenging to manage compared to other types. Example



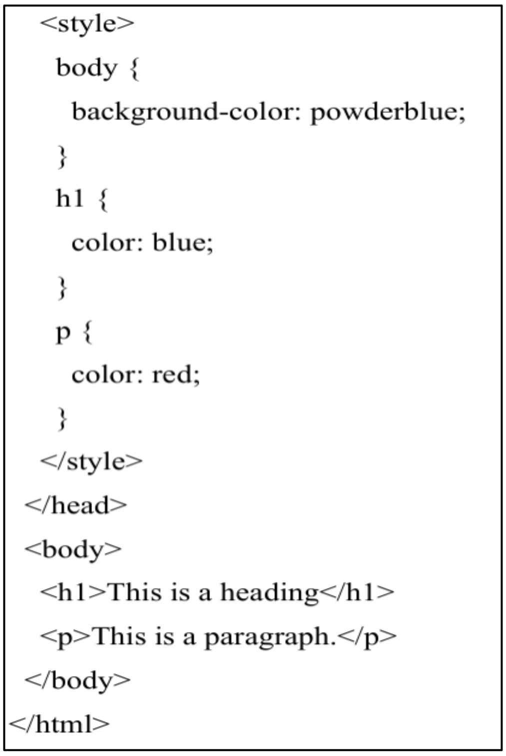
```
<!DOCTYPE html>
<html>
 <body>
  <h1 style="color: blue; text-align: center;">This is a
placeholder of heading</h1>
  <p style="color: red;">This is a placeholder for a
paragraph.</p>
 </body>
</html>
```

Figure 1.10

In this example, the style attribute is used to apply inline styles directly to the HTML elements. The color property is set to blue for the heading and red for the paragraph, and the text-align property is set to center for the heading.

2. **Internal CSS:** This type of CSS is used to style a single web page by defining styles within the section of the HTML document. It is more time-consuming and less efficient for styling multiple pages.

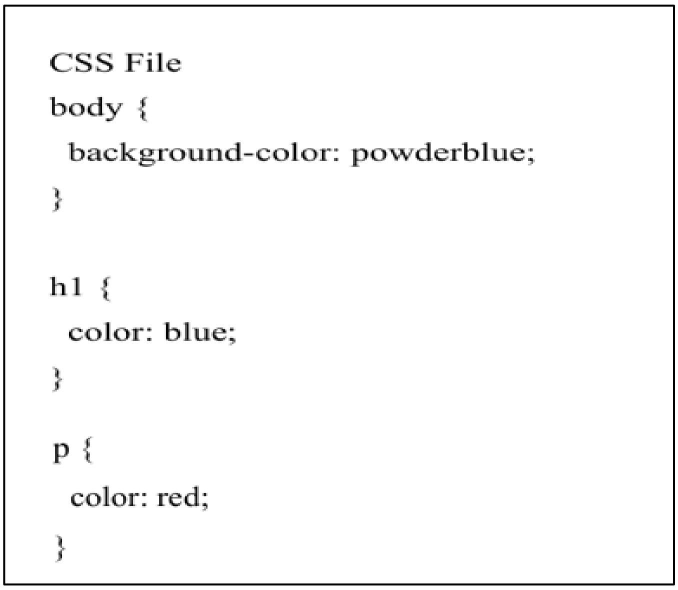
Example :



```
  <style>
   body {
    background-color: powderblue;
   }
   h1 {
    color: blue;
   }
   p {
    color: red;
   }
  </style>
 </head>
 <body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
 </body>
</html>
```

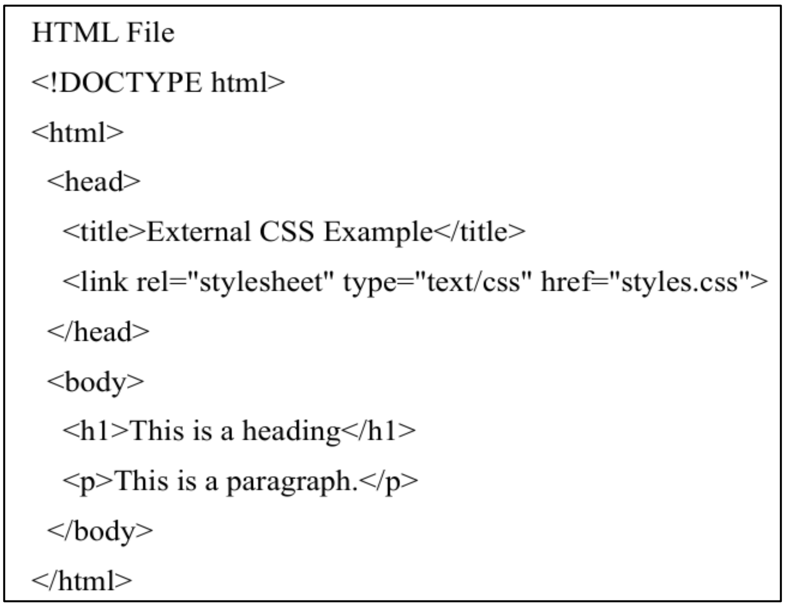
Figure 1.11

3. **External CSS:** In this type of CSS, you can modify the design of an entire website by updating a single file.. Each HTML page must include a reference to the external CSS file within the <link> element, located in the <head> section.



```
CSS File
body {
 background-color: powderblue;
}

h1 {
 color: blue;
}

p {
 color: red;
}
```

Figure 1.14



```
HTML File
<!DOCTYPE html>
<html>
  <head>
    <title>External CSS Example</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

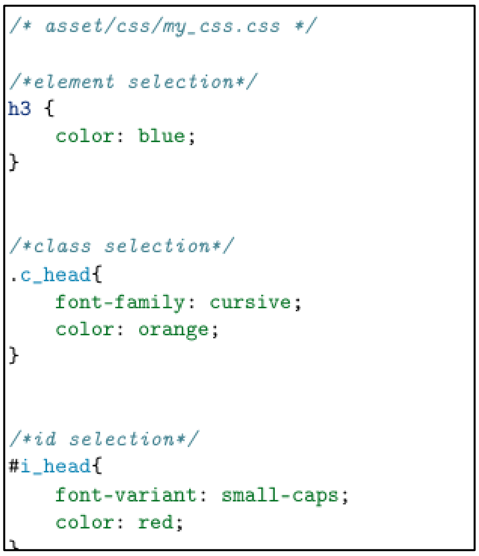
Figure 1.12

1.3.2 How to Style with CSS:

To style with CSS, we can use various selectors and properties. Here are some key concepts:

• **Selectors:** These are used to pinpoint the HTML components on web pages that need styling. There are three types of selectors in CSS

1. Element: can be selected using it's name e.g. 'p', 'div' and 'h1' etc.
2. Class: can be selected using '.className' operator e.g. '.h3_blue'.
3. ID: can be selected using '#idName' e.g. '#my_para'.



```
/* asset/css/my_css.css */

/*element selection*/
h3 {
    color: blue;
}


/*class selection*/
.c_head{
    font-family: cursive;
    color: orange;
}


/*id selection*/
#i_head{
    font-variant: small-caps;
    color: red;
}
```

Figure 1.13

• **Properties:** These define the styles to be applied to the selected HTML elements. Some common properties include color, font-size, background-color, border, and text-align.

• **Values:** These specify the values for the properties. For example, color: red sets the text color to red.

1.3.3 CONCEPT OF CLASS AND ID:

ID: It is used to identify a single, unique element on a web page.

- An ID must be unique within the HTML document - there can only be one element with a given ID.
- IDs are defined using the # symbol followed by the ID name, e.g. #my-unique-id.
- IDs have a high level of specificity, meaning they take precedence over other selectors like classes.
- IDs are commonly used to apply styles to a specific, unique element.

Class: Classes are used to group and style multiple elements on a web page.

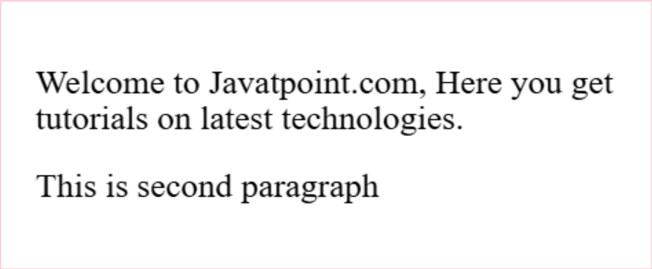
- The same class can be applied to any number of HTML elements.
- Classes are defined using the . symbol followed by the class name, e.g. .my-class.
- Classes have a lower specificity than IDs, so styles applied via a class can be overridden by styles applied via an ID.
- Classes are commonly used to apply styles to a group of elements with similar characteristics.

1.3.4 Div in CSS:

In CSS, div is a generic container that can hold any other HTML elements. It is used to group similar content together and apply styles to it. The element is a block level element

Example :

```
<div style="border:1px solid pink; padding:20px; font-size:20px">

<p>Welcome to Javatpoint.com, Here you get tutorials on latest technologies </p>

<p>This is second paragraph</p> </div>
```



Welcome to Javatpoint.com, Here you get tutorials on latest technologies.

This is second paragraph

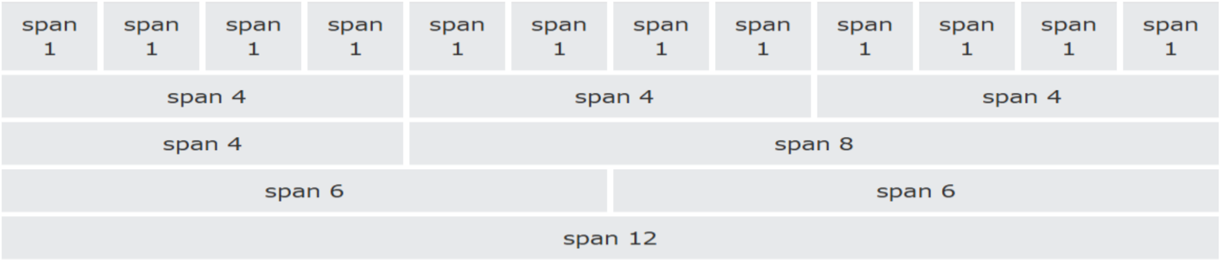
Figure 1.14

1.4 BOOTSTRAP

Bootstrap is a popular free front-end framework that streamlines web development with HTML, CSS, and JavaScript templates. It supports responsive, mobile-first designs and includes components like forms, buttons, and navigation, along with optional JavaScript plugins.

1.4.1 GRID

Bootstrap's grid system supports up to 12 columns per page, allowing for the creation of wider columns by grouping them. It is also responsive, automatically rearranging columns based on the screen size.



span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1

span 4 | span 4 | span 4

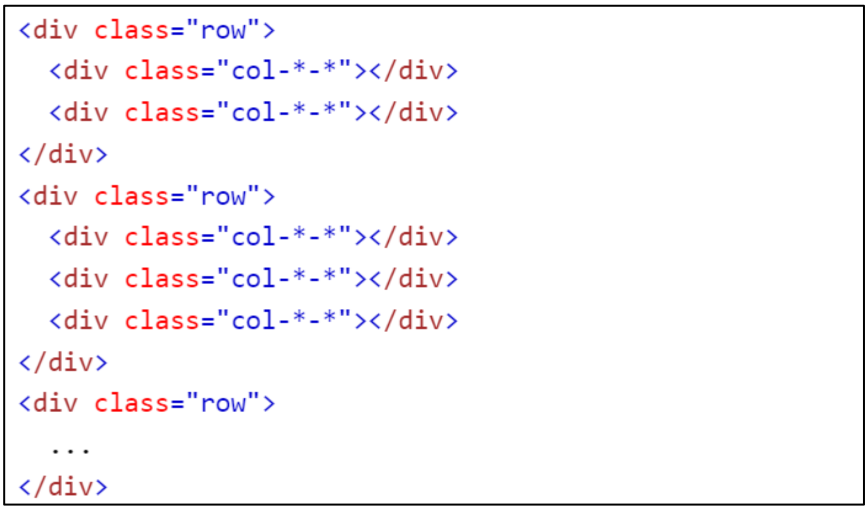
span 4 | span 8

span 6 | span 6

span 12

Figure 1.18

The following is a basic structure of a Bootstrap grid:

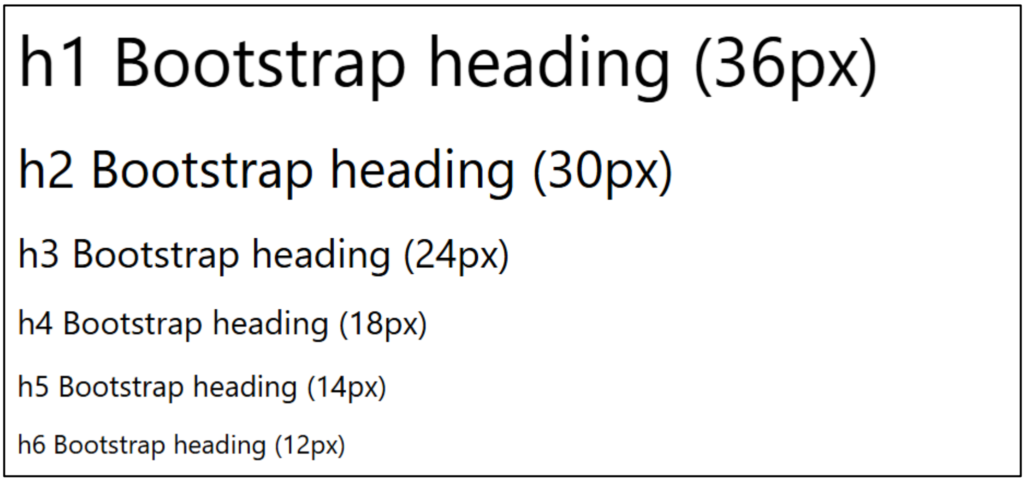


```
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  ...
</div>
```

Figure 1.19

1.4.2 Text/Typography

- **Heading Styles :** By default, Bootstrap will style the HTML headings (<h1> to <h6>) in the following way:



h1 Bootstrap heading (36px)

h2 Bootstrap heading (30px)

h3 Bootstrap heading (24px)

h4 Bootstrap heading (18px)

h5 Bootstrap heading (14px)

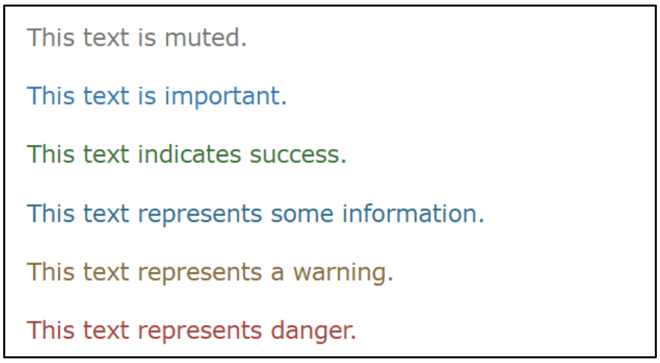
h6 Bootstrap heading (12px)

Figure 1.20

- **Contextual Colors and Backgrounds :** Bootstrap also has some contextual classes that can be used to provide "meaning through colors".

  The classes for text colors are:.text-muted, .text-primary, .text-success, .text-info, .text-warning, and .text-danger

Example :



This text is muted.

This text is important.

This text indicates success.

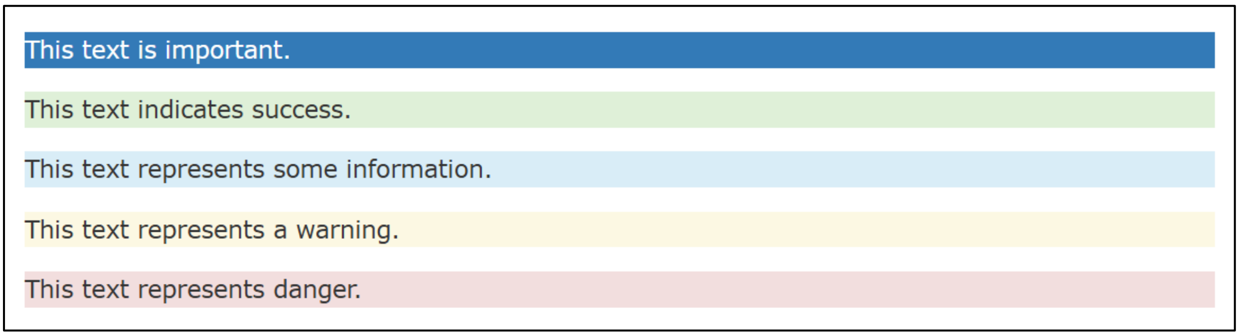
This text represents some information.

This text represents a warning.

This text represents danger.

Figure 1.21

The classes for background colors are:.bg-primary, .bg-success, .bg-info, .bg-warning, and .bg-danger



This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

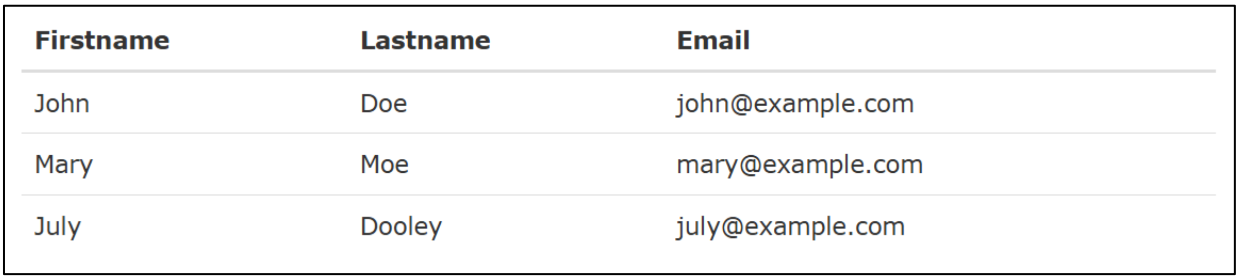
Figure 1.22

1.4.3 TABLE

A basic Bootstrap table has a light padding and only horizontal dividers.

The .table class adds basic styling to a table:

Example :



Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

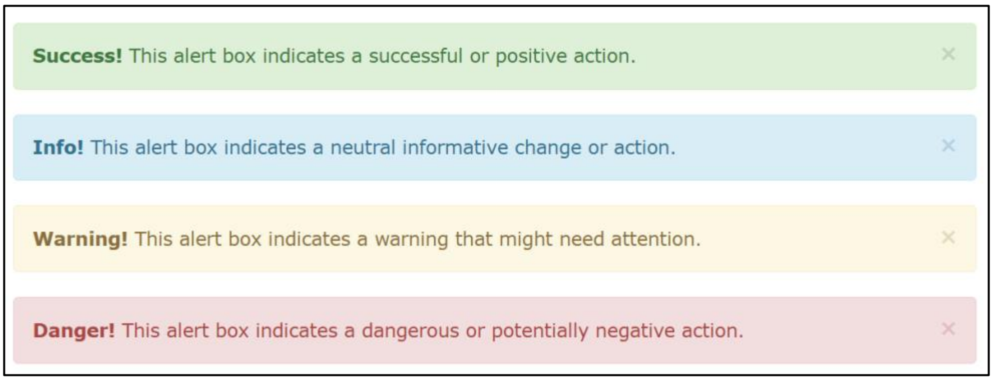
Figure 1.23

There are various types of tables in Bootstrap that can be created using different table classes, which are as follows:

1. **.table-striped**: Adds zebra stripes to rows for better readability.
2. **.table-bordered**: Adds borders to all sides of the table and cells.
3. **.table-hover**: Highlights table rows with a background color on hover.
4. **.table-condensed**: Reduces cell padding for a more compact table.
5. **.table-responsive**: Makes tables scroll horizontally on small devices.

1.4.4 ALERTS

Bootstrap provides an easy way to create predefined alert messages:



Success! This alert box indicates a successful or positive action. ×

Info! This alert box indicates a neutral informative change or action. ×

Warning! This alert box indicates a warning that might need attention. ×

Danger! This alert box indicates a dangerous or potentially negative action. ×

Figure 1.24

Alerts are created with the .alert class, followed by one of the four contextual classes .alert-success, .alert-info, .alert-warning or .alert-danger

1.4.5 BUTTONS

Bootstrap provides different styles of buttons:

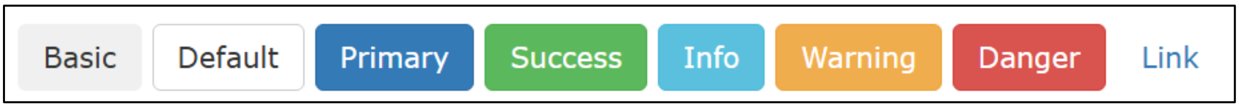
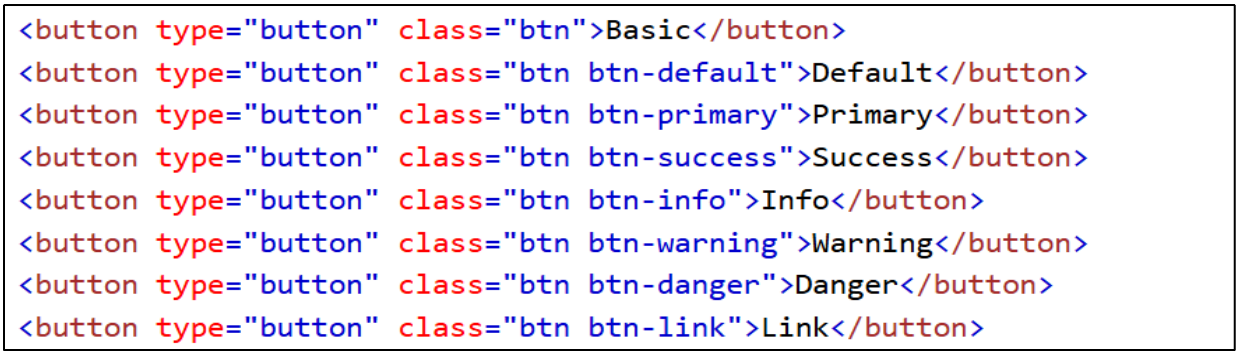


Figure 1.25

To achieve the button styles above, Bootstrap has the following classes:

- .btn
- .btn-default
- .btn-primary
- .btn-success
- .btn-info
- .btn-warning
- .btn-danger
- .btn-link

The following example shows the code for the different button styles:



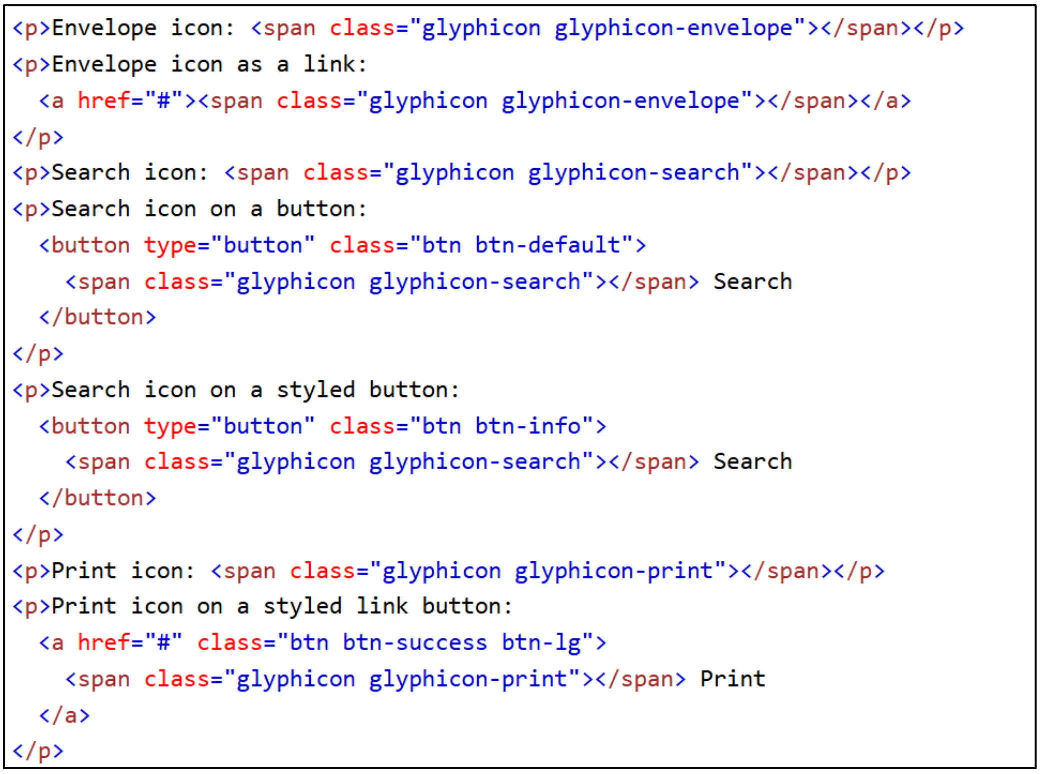
```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

Figure 1.26

1.4.6 GLYPHICONS

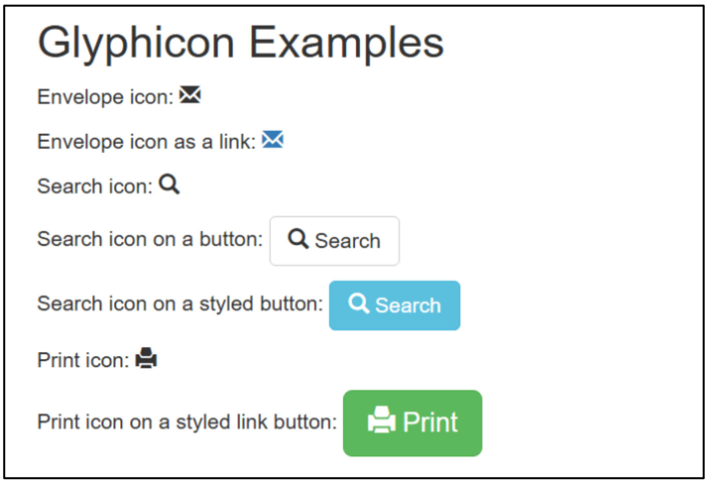
Bootstrap provides 260 glyphicons from the Glyphicons Halflings set. Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc.

Example :



```
<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
<p>Envelope icon as a link:
  <a href="#"><span class="glyphicon glyphicon-envelope"></span></a>
</p>
<p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
<p>Search icon on a button:
  <button type="button" class="btn btn-default">
    <span class="glyphicon glyphicon-search"></span> Search
  </button>
</p>
<p>Search icon on a styled button:
  <button type="button" class="btn btn-info">
    <span class="glyphicon glyphicon-search"></span> Search
  </button>
</p>
<p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
<p>Print icon on a styled link button:
  <a href="#" class="btn btn-success btn-lg">
    <span class="glyphicon glyphicon-print"></span> Print
  </a>
</p>
```

Figure 1.27



Glyphicon Examples

Envelope icon:

Envelope icon as a link:

Search icon:

Search icon on a button:

Search icon on a styled button:

Print icon:

Print icon on a styled link button:

Figure 1.28

1.4.7 Navigation Bar

A navigation bar is a navigation header that is placed at the top of the page:

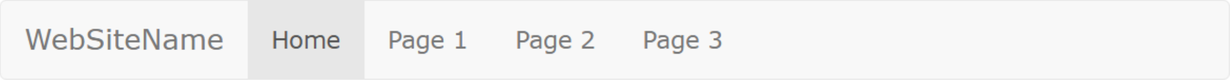


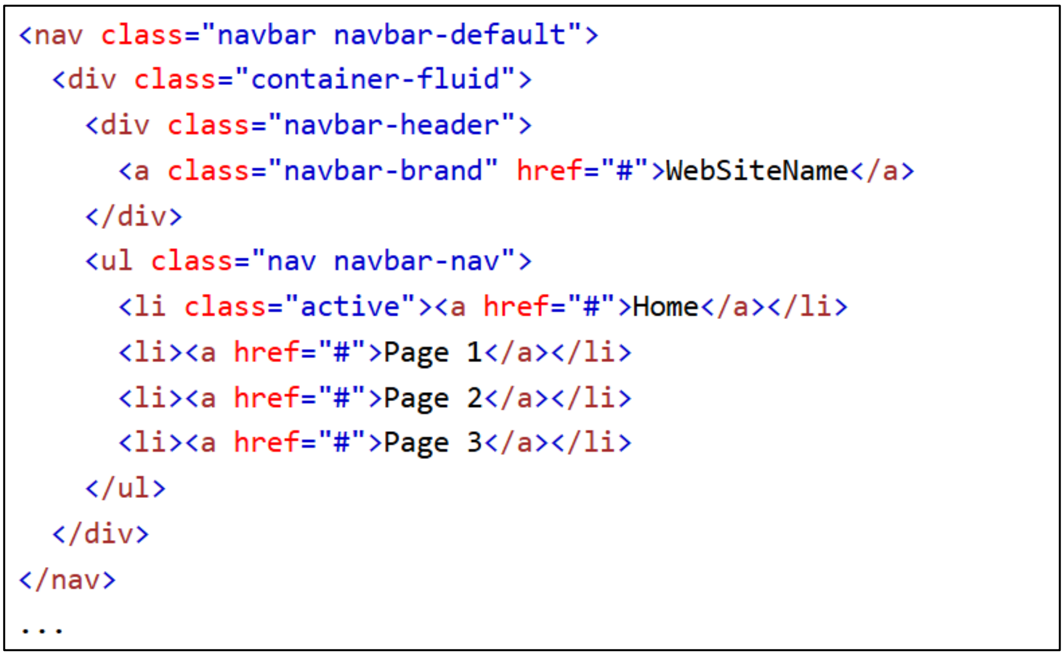
Figure 1.29

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with <nav class="navbar navbar-default">.

The following example shows how to add a navigation bar to the top of the page:

Example :



```html
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Page 1</a></li>
      <li><a href="#">Page 2</a></li>
      <li><a href="#">Page 3</a></li>
    </ul>
  </div>
</nav>
...
```

Figure 1.30

There are various types of nav bar in bootstrap

1. Standard Navigation Bar
2. Inverted Navigation Bar
3. Fixed Navigation Bar
4. Collapsing Navigation Bar

1.4.8 FORMS

Bootstrap provides three types of form layouts:

1. Vertical form (this is default)
2. Horizontal form

3. Inline form

To ensure consistent spacing and styling:

- Enclose labels and form controls within <div class="form-group"> elements.
- Apply the .form-control class to all textual <input>, <textarea>, and <select> elements.

Example (Vertical Form) :

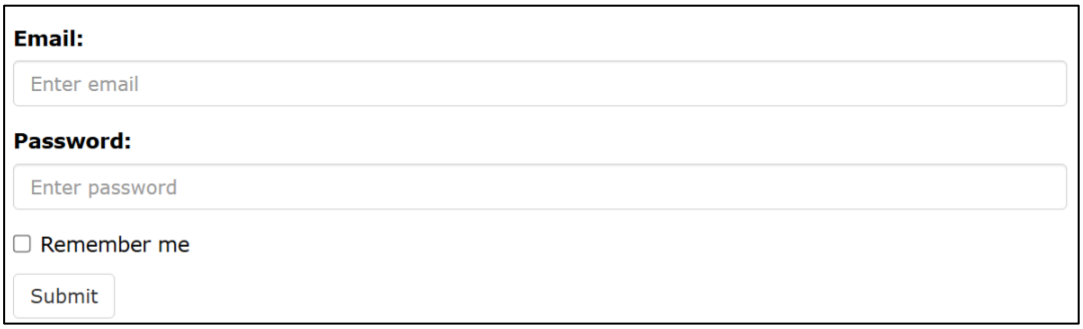
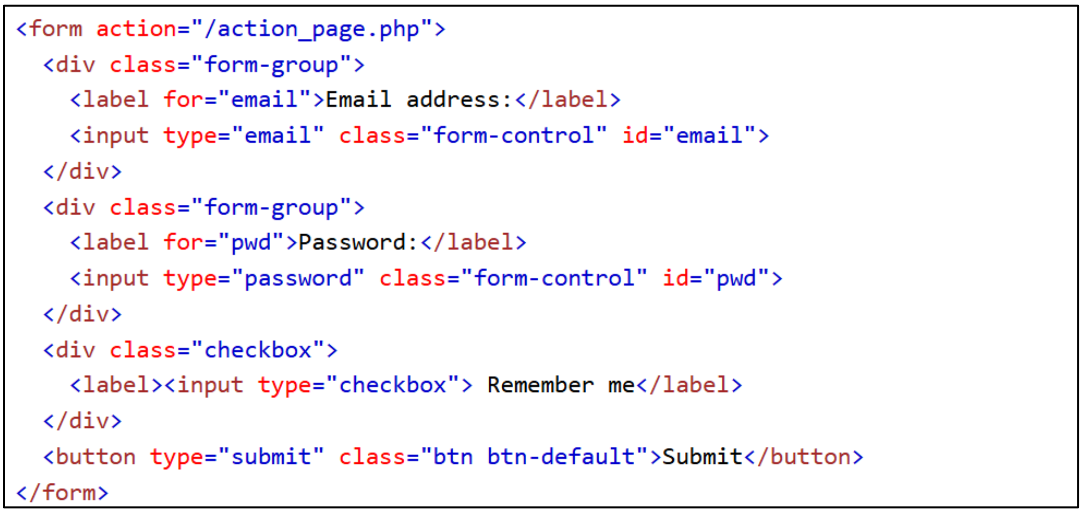


Figure 1.31



```
<form action="/action_page.php">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Figure 1.32

1.5 JAVASCRIPT

JavaScript is a high-level, dynamic programming language primarily used for enhancing interactivity and functionality on websites. It enables developers to create responsive user interfaces, control multimedia, animate images.

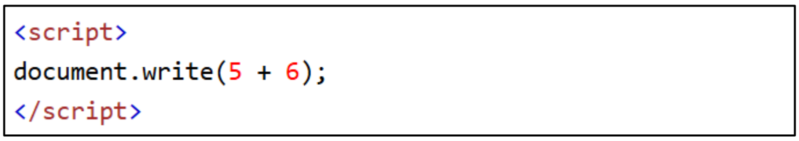
1.5.1 OUTPUT

JavaScript can "display" data in different ways:

1. **Using document.write() :**

For testing purposes, it is convenient to use document.write()

Example :



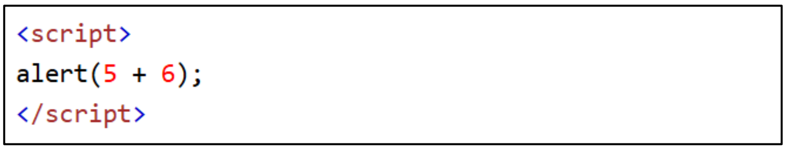
```
<script>
document.write(5 + 6);
</script>
```

Figure 1.33

2. **Using window.alert() :**

You can use an alert box to display data

Example **:**



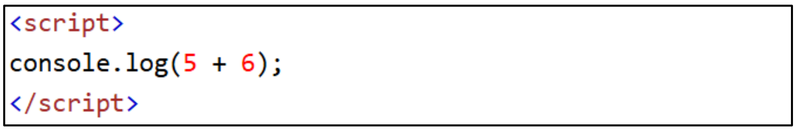
```
<script>
alert(5 + 6);
</script>
```

Figure 1.34

3. **Using console.log() :**

For debugging purposes, you can call the console.log() method in the browser to display data.

Example :



```
<script>
console.log(5 + 6);
</script>
```

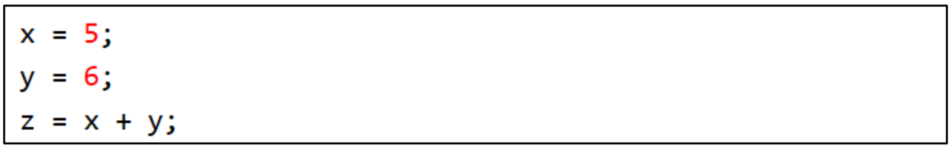
Figure 1.35

1.5.2 VARIABLES

Variables are Containers for Storing Data . JavaScript Variables can be declared in 4 ways:

1. **Automatically**

Declares global variables without keywords (not recommended). Example :



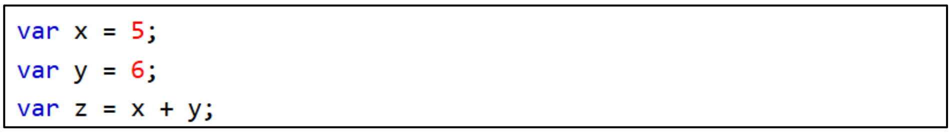
```
x = 5;
y = 6;
z = x + y;
```

Figure 1.36

2. Using var

Function-scoped, older way of declaring variables.

Example :



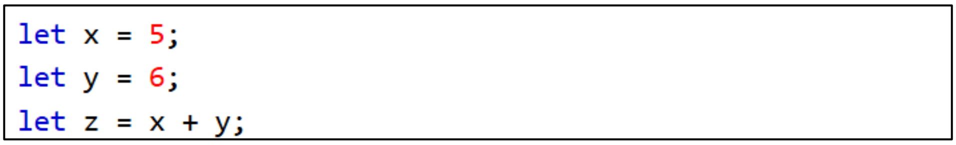
```
var x = 5;
var y = 6;
var z = x + y;
```

Figure 1.37

3. Using let

Block-scoped, allows reassignment. Safer and modern.

Example :



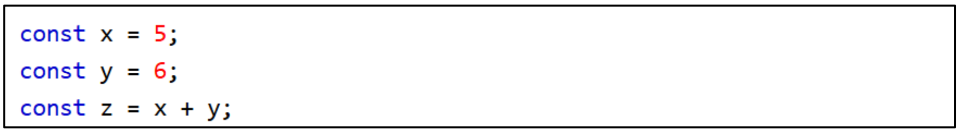
```
let x = 5;
let y = 6;
let z = x + y;
```

Figure 1.38

4. Using const

Block-scoped, cannot be reassigned. Ideal for fixed values.

Example :



```
const x = 5;
const y = 6;
const z = x + y;
```

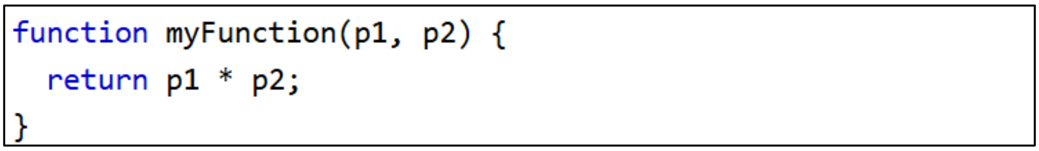
Figure 1.39

1.5.2 FUNCTIONS

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

Example :



```
function myFunction(p1, p2) {
  return p1 * p2;
}
```

Figure 1.40

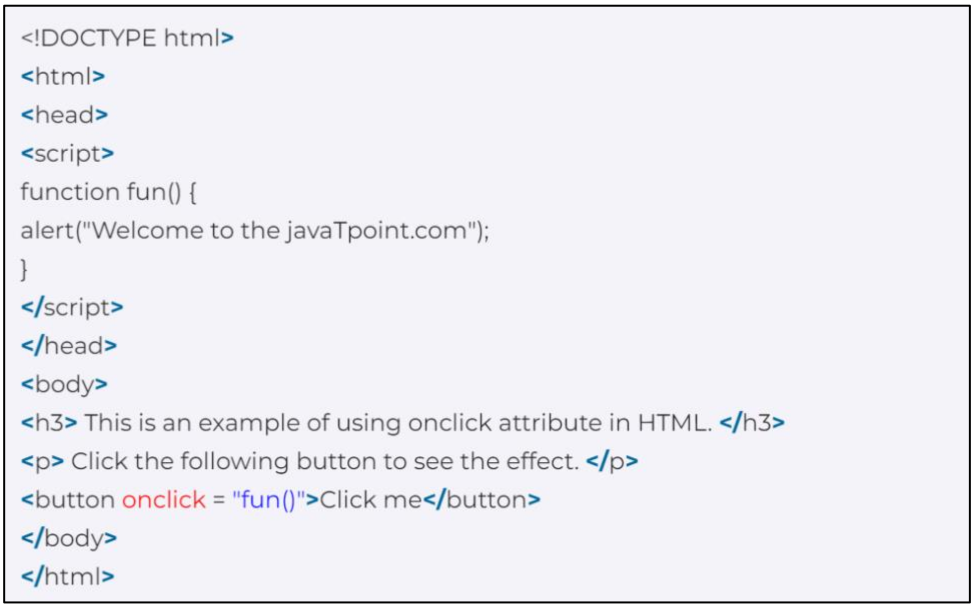
1.5.3 EVENTS

JavaScript events are actions or occurrences that happen in the browser, which JavaScript can respond to. Here are some common events:

1. onclick

Triggered when an element is clicked.

Example :



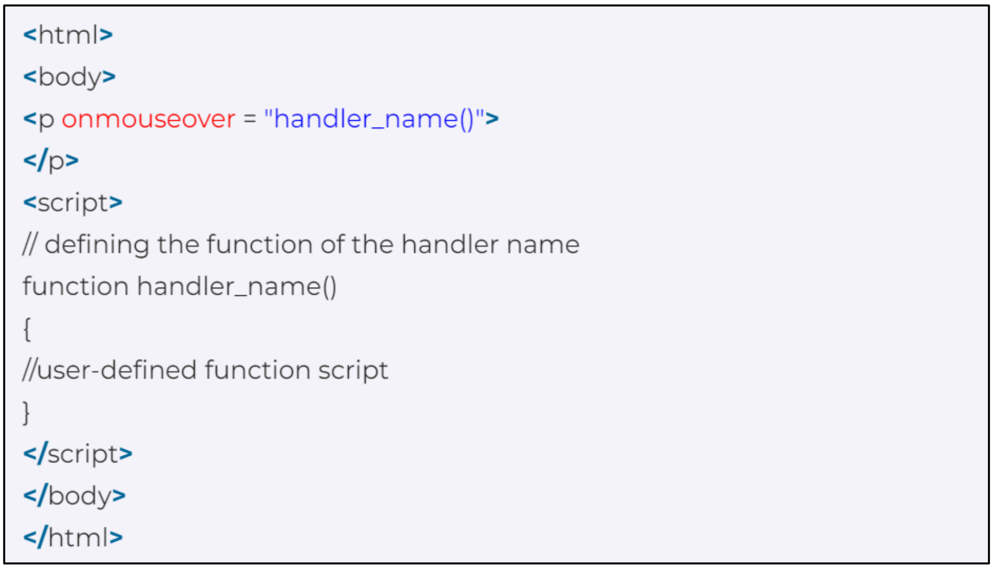
```
<!DOCTYPE html>
<html>
<head>
<script>
function fun() {
alert("Welcome to the javaTpoint.com");
}
</script>
</head>
<body>
<h3> This is an example of using onclick attribute in HTML. </h3>
<p> Click the following button to see the effect. </p>
<button onclick = "fun()">Click me</button>
</body>
</html>
```

Figure 1.41

2. mouseover

Fires when the mouse pointer moves over an element.

Example :



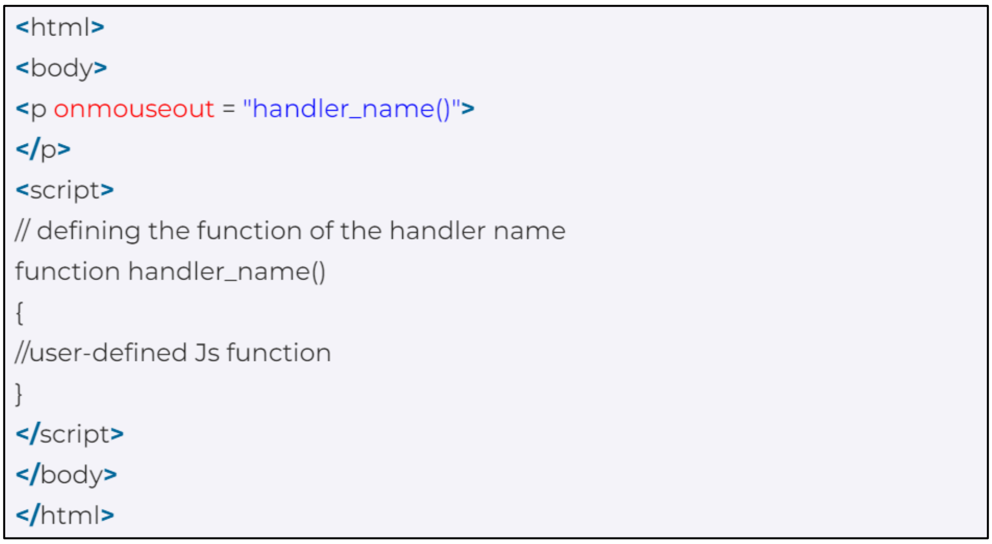
```
<html>
<body>
<p onmouseover = "handler_name()">
</p>
<script>
// defining the function of the handler name
function handler_name()
{
//user-defined function script
}
</script>
</body>
</html>
```

Figure 1.42

3. mouseout

Fires when the mouse pointer moves out of an element.

Example :



```
<html>
<body>
<p onmouseout = "handler_name()">
</p>
<script>
// defining the function of the handler name
function handler_name()
{
//user-defined Js function
}
</script>
</body>
</html>
```

Figure 1.43

CHAPTER-2 TRAINING WORK UNDERTAKEN

2.1 Sequential Learning Steps

2.1.1 HTML Fundamentals

Introduced to the basics of HTML, including document structure, tags, and attributes. Practiced creating headings, paragraphs, links, images, and lists. Worked with HTML tables for organizing data in rows and columns and built forms for user input collection.

2.1.2 CSS Styling

Learned CSS basics, including syntax, selectors, and properties. Applied CSS to style elements and control appearance and layout. Created responsive layouts using techniques like flexbox and grid to ensure adaptability to different screen sizes.

2.1.3 BOOTSTRAP

Learned to create responsive layouts using a 12-column grid system. Customized typography with different text styles, sizes, and weights. Created responsive tables with various styles and displayed informative alerts. Styled buttons in various sizes and colors and enhanced the user interface using glyphicons. Built and styled navigation bars with different configurations and forms with various input types and validation. Designed and implemented a complete Bootstrap template.

2.1.4 JavaScript Fundamentals

Understood JavaScript basics, including variables, data types, operators, and control flow. Created reusable functions and worked with objects. Handled events to respond to user interactions such as clicks and hovers.

2.1.5 Frontend Project

Planned the project by defining requirements and creating a project plan. Developed the frontend by designing and implementing the user interface using HTML, CSS, Bootstrap, and JavaScript.

2.2 Methodology Followed

2.2.1 Instructional Method

The training program combined lectures, hands-on exercises, and project-based learning to teach web development skills. Key topics included HTML, CSS, Bootstrap and JavaScript . Trainees applied their knowledge through projects and received guidance from experienced mentors.

2.2.2 Assessment and Feedback

To evaluate the training program's effectiveness and trainee progress, continuous assessment methods were used, including regular quizzes and assignments with feedback for improvement. The final project was evaluated for functionality, design, and best practices, with feedback on strengths and weaknesses. Peer review encouraged trainees to assess each other's work, fostering critical thinking and communication skills. These assessment techniques aimed to create a comprehensive and engaging learning experience.

2.3 Project Undertaken

A responsive website was developed using HTML, CSS, Javscript and Bootstrap to showcase a school website.

2.3.1 Key Features

1. **Intuitive Navigation**: A clear and user-friendly navigation bar guides visitors through the website's sections.
2. **Engaging Homepage**: A visually appealing homepage highlights the school's mission, vision, and unique selling points.
3. **Informative About Us Page**: Provides detailed information about the school's history, philosophy, and faculty.
4. **Comprehensive Admissions Page**: Offers information about the admission process, eligibility criteria, and important dates.

5. **Captivating Gallery**: Showcases the school's facilities, events, and student activities through a visually appealing gallery.
6. **Contact Us Page**: Provides contact information and a contact form for inquiries.
7. **Responsive Design**: Ensures optimal viewing experience across various devices (desktop, tablet, mobile).

2.3.2 Technical Skills Employed

1. **HTML**: Structure and content organization.
2. **CSS**: Styling and layout.
3. **Bootstrap**: Framework for rapid and responsive web development.

By leveraging Bootstrap's pre-built components and grid system, the website was developed efficiently and effectively.

CHAPTER-3 RESULT AND DISCUSSION

This chapter outlines the outcomes of the one-month web development training program at HB Infotech Solutions and discusses the knowledge and skills acquired during the training. It also highlights the practical application of these skills through a web development project:

3.1 KEY OUTCOMES

3.1.1 HTML :

- Mastered HTML structure, tags, and syntax.
- Created forms for user input and tables for data display.
- Used semantic tags for better accessibility and SEO.

3.1.2 CSS :

- Learned CSS syntax and layout techniques (Flexbox, Grid).
- Styled typography and optimized designs for responsiveness with media queries.
- Applied visual enhancements like shadows, gradients, and transitions.

3.1.3 BOOTSRAP :

- Used Bootstrap's 12-column grid for responsive layouts.
- Implemented components like navigation bars, buttons, and forms.
- Customized themes and created responsive tables.

3.1.4 JAVASCRIPT :

- Gained a solid understanding of **JavaScript syntax**, including variables, operators, data types, and functions.
- Handled **user interactions** using events like **onclick**, **onmouseover**, and **onmouseout**.
- Applied **basic control flow** (if-else, loops)

3.2 Project Demonstration

To demonstrate the practical application of the acquired skills, a web application was developed during the training period. The application showcased the following features:

- **Intuitive Navigation**: A clear and simple navigation structure for easy exploration.
- **Engaging Visuals**: Attractive design elements for an appealing user experience.
- **Comprehensive Content**: Key sections providing relevant and detailed information.
- **Interactive Features**: A gallery and contact form for user engagement.
- **Responsive Design**: Optimized for smooth performance across different devices and screen sizes.

3.3 Discussion

The one-month training program provided a solid foundation in **web development**, focusing on **HTML**, **CSS**, **JavaScript**, and **Bootstrap**. Through a hands-on project, the trainee developed a functional, responsive web application. While proficiency in **JavaScript** and **Bootstrap** is still developing, the training provided a strong understanding of **JavaScript fundamentals** like **event handling** and **control flow**, and introduced **Bootstrap** for creating responsive layouts quickly. Further practice is needed to master both, but these tools proved effective for building the project efficiently.

The project helped integrate these skills, solidifying the trainee's understanding of **front-end development** and how to use these technologies together to create a seamless user experience. This practical experience has set the stage for continued learning, focusing on more advanced **JavaScript** and **Bootstrap** techniques. The training program successfully equipped the trainee with the knowledge and skills to build responsive websites, laying a foundation for further growth in web development.

CHAPTER-4 CONCLUSION AND FUTURE SCOPE

4.1 CONCLUSION

The training in front-end web development has provided you with essential skills in HTML, CSS, and JavaScript, enabling you to build interactive, responsive, and visually appealing user interfaces. Through practical projects, I have gained experience in creating web pages that prioritize user experience and accessibility, preparing you to design effective, user-centered websites. This training has laid a strong foundation in modern web practices, positioning you well to contribute to real-world projects and collaborate with design and development teams.

4.2 FUTURE SCOPE

- **Progressive Web Apps (PWAs):** Apps that combine the web's reach with mobile app functionality, making experiences faster and offline-capable.
- **Advanced JavaScript Frameworks**: Mastering frameworks can open up opportunities to work on complex, scalable front-end applications .
- **Focus on Accessibility**: Skills in accessible design will help you build more inclusive websites, a growing priority for companies.
- **User Interface and Experience (UI/UX) Design**: Enhancing your understanding of UI/UX principles and design tools (like Figma or Adobe XD) can improve your ability to create intuitive, user-friendly interfaces.
- **Cross-Platform Development**: Familiarity with frameworks like React Native can enable you to extend your skills to mobile app development, broadening your career options.

The future of web development will focus on faster, safer, and more personalized digital experiences.